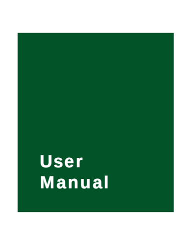


XGate-COP10

CANopen 嵌入式通信模块

UM01010101　V1.01　Date: 2009/10/15　产品用户手册



类别	内容
关键词	CANopen UART 嵌入式通信模块　XGate XGate-COP10 串口通信　CANopen 从站　CAN-bus
摘　要	

广州致远电子有限公司



修订历史

版本	日期	原因
V1.00	2009/10/15	创建文档
V1.01	2010/1/22	增中通信串口波特更改命令

销售与服务网络（一）

广州周立功单片机发展有限公司

地址：广州市天河北路 689 号光大银行大厦 12 楼 F4
邮编：510630
电话：(020)38730916 38730917 38730972 38730976 38730977
传真：(020)38730925
网址：www.zlgmcu.com

广州专卖店

地址：广州市天河区新赛格电子城 203-204 室
电话：(020)87578634 87569917
传真：(020)87578842

南京周立功

地址：南京市珠江路 280 号珠江大厦 2006 室
电话：(025)83613221 83613271 83603500
传真：(025)83613271

北京周立功

地址：北京市海淀区知春路 113 号银网中心 A 座 1207-1208 室（中发电子市场斜对面）
电话：(010)62536178 62536179 82628073
传真：(010)82614433

重庆周立功

地址：重庆市石桥铺科园一路二号大西洋国际大厦（赛格电子市场）1611 室
电话：(023)68796438 68796439
传真：(023)68796439

杭州周立功

地址：杭州市天目山路 217 号江南电子大厦 502 室
电话：(0571)28139611 28139612 28139613 28139615 28139616 28139618
传真：(0571)28139621

成都周立功

地址：成都市一环路南二段 1 号数码同人港 401 室（磨子桥立交西北角）
电话：(028)85439836 85437446
传真：(028)85437896

深圳周立功

地址：深圳市深南中路 2070 号电子科技大厦 C 座 4 楼 D 室
电话：(0755)83781788（5 线）
传真：(0755)83793285

武汉周立功

地址：武汉市洪山区广埠屯珞瑜路 158 号 12128 室（华中电脑数码市场）
电话：(027)87168497 87168297 87168397
传真：(027)87163755

上海周立功

地址：上海市北京东路 668 号科技京城东座 7E 室
电话：(021)53083452 53083453 53083496
传真：(021)53083491

西安办事处

地址：西安市长安北路 54 号太平洋大厦 1201 室
电话：(029)87881296 83063000 87881295
传真：(029)87880865

销售与服务网络（二）

广州致远电子有限公司



地址：广州市天河区车陂路黄洲工业区 3 栋 2 楼
邮编：510660
传真：(020)38601859
网址：www.embedtools.com （嵌入式系统事业部）
www.embedcontrol.com （工控网络事业部）
www.ecardsys.com （楼宇自动化事业部）

技术支持：

CAN-bus：

电话：(020)22644381 22644382 22644253
邮箱：can.support@embedcontrol.com

iCAN 及数据采集：

电话：(020)28872344 22644373
邮箱：ican@embedcontrol.com

MiniARM：

电话：(020)28872684 28267813
邮箱：miniarm.support@embedtools.com

以太网：

电话：(020)22644380 22644385
邮箱：ethernet.support@embedcontrol.com

无线通讯：

电话：(020) 22644386
邮箱：wireless@embedcontrol.com

串行通讯：

电话：(020)28267800 22644385
邮箱：serial@embedcontrol.com

编程器：

电话：(020)22644371
邮箱：programmer@embedtools.com

分析仪器：

电话：(020)22644375 28872624 28872345
邮箱：tools@embedtools.com

ARM 嵌入式系统：

电话：(020)28872347 28872377 22644383 22644384
邮箱：arm.support@zlgmcu.com

楼宇自动化：

电话：(020)22644376 22644389 28267806
邮箱：mjs.support@ecardsys.com
mifare.support@zlgmcu.com

销售：

电话：(020)22644249 22644399 22644372 22644261 28872524
28872342 28872349 28872569 28872573 38601786

维修：

电话：(020)22644245

名词解释：

PDO：Process Data Object，过程数据对象。

TPDO：Transmit Process Data Object，发送过程数据对象。

RPDO：Receive Process Data Object，接收过程数据对象。

SDO：Service Data Object，服务数据对象。

NMT：Network Management，网络管理。

SYNC: Synchronization Objects，同步报文对象。

EMCY: Emergency Objects，紧急对象报文。

CAN-ID: Controller Area Network-Identify,控制器局域网标识符。

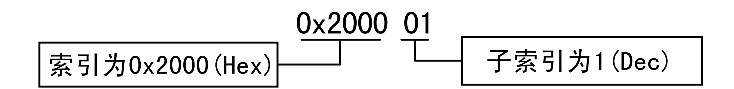
COB-ID: Communication Object-Identify,通信对象标识符。

SSDO：Servers Service Data Object，服务数据服务器。

DS：Draft Standard,标准草案。

书写规则：

本手册中，对象字典索引与子索引的书写遵循如下图所示的规则，其中索引为16进制表示，子索引为10进进制表示，索引与子索引中间用空格隔开。



目 录

1. CANopen协议简介

CANopen 协议是在 20 世纪 90 年代末，由 CiA 组织（CAN-in-Automation）在 CAL（CAN Application Layer）的基础上发展而来，一经推出便在欧洲得到了广泛的认可与应用。经过对 CANopen 协议规范文本的多次修改，使得 CANopen 协议的稳定性、实时性、抗干扰性都得到了进一步的提高。并且 CiA 在各个行业不断推出设备子协议，使 CANopen 协议在各个行业得到更快的发展与推广。目前 CANopen 协议已经在运动控制、车辆工业、电机驱动、工程机械、船舶海运等行业得到广泛的应用。

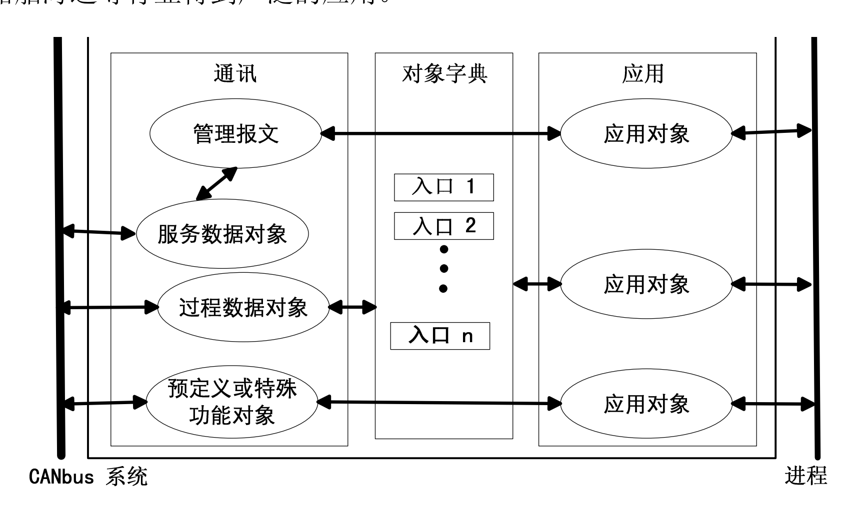


图 1　CANopen 设备结构

图 1所示为CANopen设备结构，CANopen协议通常分为用户应用层、对象字典、以及通讯三个部分。

1.1　CANopen对象字典

CANopen 对象字典(OD: Object Dictionary)是 CANopen 协议最为核心的概念。所谓的对象字典就是一个有序的对象组，每个对象采用一个 16 位的索引值来寻址，这个索引值通常被称为索引，其有效范围在 0x1000 到 0x9FFF 之间。为了允许访问数据结构中的单个元素，同时也定义了一个 8 位的索引值，这个索引值通常被称为子索引。

每个 CANopen 设备都有一个对象字典，对象字典包含了描述这个设备和它的网络行为的所有参数，对象字典通常用电子数据文档（EDS：Electronic Data Sheet）来记录这些参数，而不需要把这些参数记录在纸上。对于 CANopen 网络中的主节点来说，不需要对 CANopen 从节点的每个对象字典项都访问。

CANopen 对象字典中的项由一系列子协议来描述。子协议为对象字典中的每个对象都描述了它的功能、名字、索引、子索引、数据类型，以及这个对象是否必需、读写属性等等，这样可保证不同厂商的同类型设备兼容。

CANopen 协议的核心描述子协议是 DS301，其包括了 CANopen 协议应用层及通信结构描述，其它的子协议都是对 DS301 协议描述文本的补充与扩展。在不同的应用行业都会起草一份 CANopen 设备子协议，子协议编号一般是 DS4xx 或 DSP4xx。

CANopen 协议包含了许多的子协议，其主要划分为以下三类。

1. 通讯子协议（Communication Profile）

通讯子协议，描述对象字典的主要形式和对象字典中的通讯对象以及参数。这个子协议适用所有的 CANopen 设备，其索引值范围从 0x1000~0x1FFF。

2. 制造商自定义子协议（Manufacturer-specific Profile）

制造商自定义子协议，对于在设备子协议中未定义的特殊功能，制造商可以在此区域根据需求定义对象字典对象。因此这个区域对于不同的厂商来说，相同的索引的对象字典项定义不一定相同，其索引值范围为 0x2000~0x5FFF。

3. 设备子协议(Device Profile)

设备子协议，为各种不同类型的设备定义对象字典中的对象。目前已有十几种为不同类型的设备定义的子协议，例如 DS401、DS402、DS406 等，其索引值范围为 0x6000~0x9FFF。

1.2 CANopen通讯

在 CANopen 协议中主要定义了管理报文对象 NMT（Network Management）、服务数据对象 SDO(Service Data Object)、过程数据对象 PDO(Process Data Object)、预定义报文或特殊功能对象等四种对象。

1. 网络管理NMT（Network Management）

管理报文负责层管理、网络管理和ID分配服务，例如，初始化、配置和网络管理（其中包括节点保护）。网络管理中，同一个网络中只允许有一个主节点、一个或多个从节点，并遵循主从模式。通过NMT 服务，我们可以对节点进行初始化、运行、监控、复位和停止。所有节点都被认为是NMT 从站。

2. 服务数据对象SDO（Service Data Object）

SDO 主要用于主节点对从节点的参数配置。服务确认是 SDO 的最大的特点，为每个消息都生成一个应答，确保数据传输的准确性。在一个 CANopen 系统中，通常 CANopen 从节点作为 SDO 服务器，CANopen 主节点作为客户端。客户端通过索引和子索引，能够访问数据服务器上的对象字典。这样 CANopen 主节点可以访问从节点的任意对象字典项的参数，并且 SDO 也可以传输任何长度的数据（当数据长度超过 4 个字节时就拆分成多个报文来传输）。

3. 过程数据对象PDO（Process Data Object）

PDO用来传输实时数据，其传输模型为生产者消费者模型如图 2所示。数据长度被限制为 1~8 字节。PDO通信对象具有如下的特点：

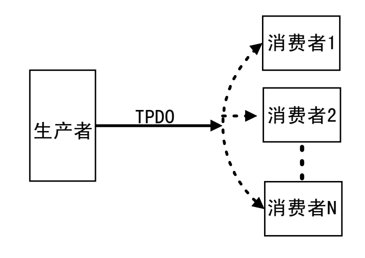


图 2 生产者消费者模型

- PDO 通讯没有协议规定，PDO 数据内容由它的 CAN-ID（也可称为 COB-ID）定义；
- 每个 PDO 在对象字典中用 2 个对象描述：
  - PDO通讯参数，该通讯参数定义了设备所使用的COB-ID、传输类型、定时周期；
  - PDO映射参数，映射参数包含了一个对象字典中的对象列表，这些对象映射到

相应的PDO，其中包括数据的长度（单位：位），对于生产者和消费者都必须要知道这个映射参数，才能够正确的解释PDO内容。

- PDO 消息内容是预定义的，如果 PDO 支持可变 PDO 映射，那么该 PDO 是可以通过 SDO 进行配置；
- PDO 可以有多种的传输方式：
  - 同步传输（通过接收同步对象实现同步），同步传输又可分为非周期和周期传输。非周期传输是由远程帧预触发或者由设备子协议中规定的对象特定事件预触发传送。周期传输则是通过接收同步对象（SYNC）来实现，可以设置1~240个同步对象触发；
  - 异步传输（由特定事件触发），其触发方式可有两种，第一种是通过发送与PDO的COB-ID相同的远程帧来触发PDO的发送，第二种是由设备子协议中规定的对象特定事件来触发（例如，定时传输，数据状态变化传输等）。

4. 预定义报文或特殊功能对象

预定义报文或特殊功能对象为 CANopen 设备提供特定的功能，方便 CANopen 主站对从站管理。在 CANopen 协议中，已经为特殊的功能预定义了 COB-ID，其主要有以下几种特殊报文：

- 同步（SYNC），该报文对象主要实现整个网络的同步传输，每个节点都以该同步报文作为 PDO 同步触发参数，因此该同步报文的 COB-ID 具有比较高的优先级以及最短的传输时间；
- 时间标记对象（Time Stamp），为各个节点提供公共的时间参考；
- 紧急事件对象（Emergency），当设备内部发生错误触发该对象，即发送设备内部错误代码；
- 节点/寿命保护（Node/Life Guarding），主节点可通过节点保护方式获取从节点的状态。从节点可通过寿命保护方式获取主节点的状态；
- 启动报文对象（Boot-up），从节点初始化完成后向网络中发送该对象，并进入到预操作状态。

1.3 CANopen网络配置

在CANopen协议描述文本DS305中定义了一种网络配置协议即网络配置服务 LSS (Layer Setting Service)，其通过CAN总线，用具有LSS 主机功能的CANOpen模块来查询或修改具有LSS 从机的CANOpen模块的某些参数。

通过使用LSS ，可以对下面的参数进行查询或修改：

- CANopen 从站的 Node-ID；
- 物理层的位定时参数 (波特率)；
- LSS 地址 (特征对象 1018h)。

2. XGate-COP10 特点

XGate-COP10为CANopen从站通信模块，其内部已经集成了CANopen从站协议栈代码，不需要进行二次开发。协议栈遵循CANopen协议描述文档DS301、DS302、DS303以及DS305标准。在默认情况下，CANopen从站启用预定义连接报文，该模块具有如下所示特点。

- 网络管理服务对象(NMT：Boot up，Node Guarding /Life guarding，Heartbeat Producer)；
- 过程数据对象(TPDO 与 RPDO)；
- 服务数据对象(SDO 服务器/SSDO)；
- 紧急报文对象(Emergency)；
- 同步报文对象(Sync)；
- 网络配置对象(LSS 从站)；
- 串口通信能力(UART,115200 bps)；
- 96 字节的输入输出数据缓冲（I/O)；
- CAN 总线支持 5Kbps～1Mbps 的波特率；
- 小体积，18mm×31mm（DIP24 封装）。

XGate-COP10从站具有高实时性，并可以支持多达12个RPDO和TPDO过程数据传输，适用于各种干扰强、实时性要求高的场合，小巧的体积（DIP24封装：宽1.8cm，长3.1cm）适用于嵌入到各种电路板中。

如图 3所示为XGate-COP10系统结构图，将用户CPU串口发送过来的数据以PDO报文方式发送到CAN总线，以及读取来自CAN总线的RPDO数据。当然也可以通过串口配置模块相关参数。

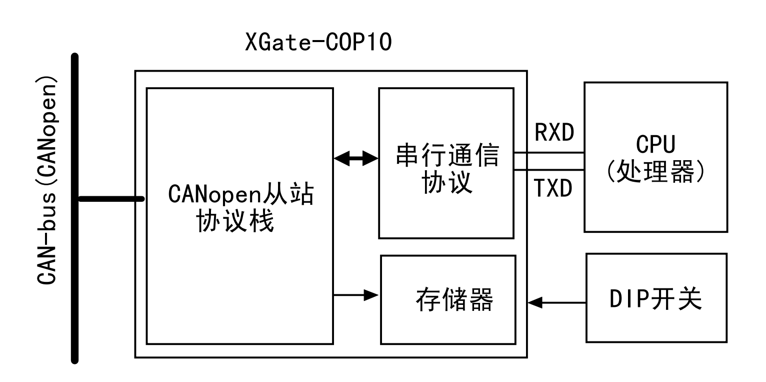


图 3　XGate-COP10 系统结构

如图 4所示为XGate-COP10和状态之间有的转换图，图中各字母所表示的各种状态下可进行的操作，字母表达的操作为：

- a. NMT
- b. Node Guard
- c. SDO
- d. Emergency
- e. PDO
- f. Boot-up。

图中箭头所表示各个状态之间的转换关系，数字表示这种转换所需要进行的操作，数字表达的操作为：

- 1: Start_Remote_node (0x01)
- 2: Stop_Remote_Node (0x02)
- 3: Enter_Pre-Operational_State (0x80)
- 4: Reset_Node (0x81)
- 5: Reset_Communication (0x82)
- 6: 设备初始化结束，自动进入 Pre_Operational 状态，发送 Boot-up 消息

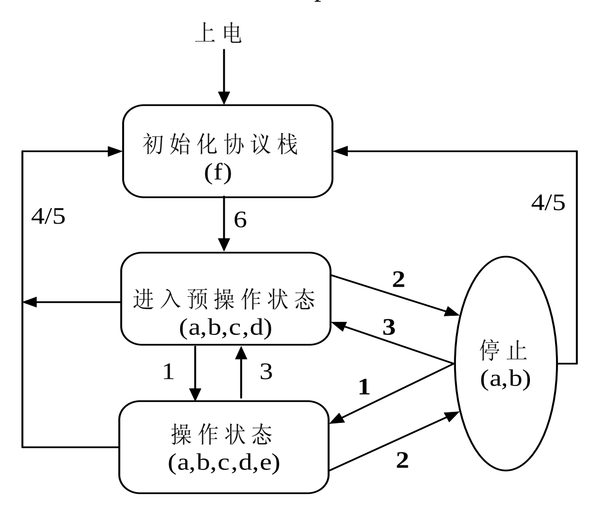


图 4 XGate-COP10 各状态转换图

2.1 XGate-COP10 状态指示灯(DS303-3)

按照CANopen协议规范文档DS303-3 的定义，在XGate-COP10 模块中使用两个LED指示灯来指示当前模块所处的状态，如表 1所示。

表 1 状态指示灯

指示灯名称	颜色	管脚
运行指示灯（RUN）	绿色	13
错误指示灯（ERR）	红色	14

其状态指示灯所指的各种状态所指示的含义如下表 2、表 3、表 4所示。

表 2 LED 状态说明

指示灯状态	现象描述
亮（LED on）	常亮
暗（LED off）	常暗
闪烁（LED flickering）	亮和暗的时间等长，频率大概是 10Hz：亮大约 50ms，暗大约 50ms
闪烁（LED blinking）	亮和暗的时间等长，频率大概是 2.5Hz：亮大约 200ms，暗大约 200ms
闪一下（LED single flash）	一个很短的闪光（大约 200ms）接着是长时间的暗（大约 1000ms）
闪两下（LED double flash）	两个很短的闪光（大约 200ms）中间用一个大约 200ms 的暗来分隔。这个序列用一个长时间的暗（大约 1000ms）来结束
闪三下（LED triple flash）	三个很短的闪光（大约 200ms）中间用大约 200ms 的暗来分隔。这个序列用一个长时间的暗（大约 1000ms）来结束

表 3 错误状态指示灯（ERR）描述

编号	ERROR LED	状态	描述	种类
1	暗	没有错误	器件处于工作状态	强制
2	闪一下	到达警戒值	CAN 控制器的至少一个错误计数器到达或超出了警戒值（错误帧太多）	强制
3	闪烁 （Flickering）	自动波特率 / LSS	正在进行自动波特率检测或进行 LSS 服务（和 RUN LED 交替闪烁（flickering）	可选
4	闪两下	错误控制事件	发生保护事件（NMT 从机或 NMT 主机）或心跳事件（心跳使用者）	强制
5	闪三下	Sync 错误	SYNC 报文超出配置的通讯循环间隔仍未收到（见对象字典条项 0x1006）	有条件，如果支持对象 0x1006 则强制
6	亮	总线关闭	CAN 控制器总线关闭	强制

表 4　运行状态指示灯（RNU）描述

编号	RUN LED	状态	描述	种类
1	Flickering（闪烁）	自动波特率 / LSS	正在进行 LSS 服务	可选
2	闪一下	停止	器件处于停止状态	强制
3	闪烁（Blinking）	预操作	器件处于预操作状态	强制
4	亮	工作	器件处于工作状态	强制
5	暗	故障	请检查模块复位引脚以及电源是否连接正确	

3. XGate-COP10 节点ID与CAN总线波特率

3.1 CANopen节点地址和波特率设置

XGate-COP10 提供三种方式设置节点ID和节点波特率，其设置顺序如图 5和图 6所示。如果用户所使用的拨码开关值有效，则上电时启用该值，即使在存储器中存储有合法的ID值都不会使用。如果模块在运行期间，主站对模块进行了LSS设置，则使用LSS设置后的值，但是模块重新上电或重启之后，依然使用拨码开关的值。

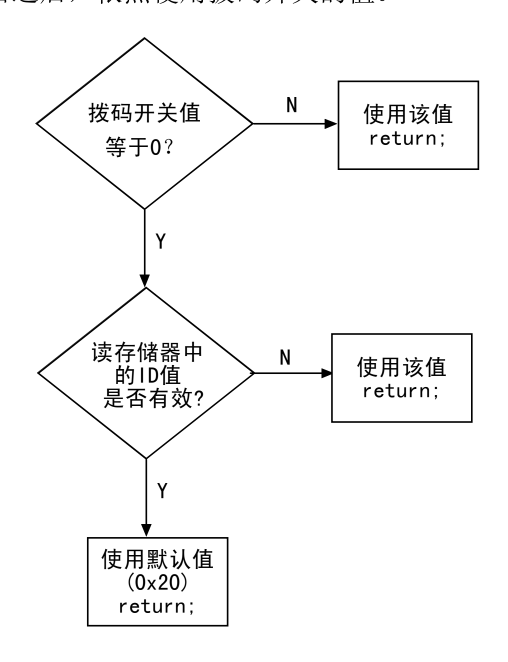


图 5　节点 ID 设置顺序

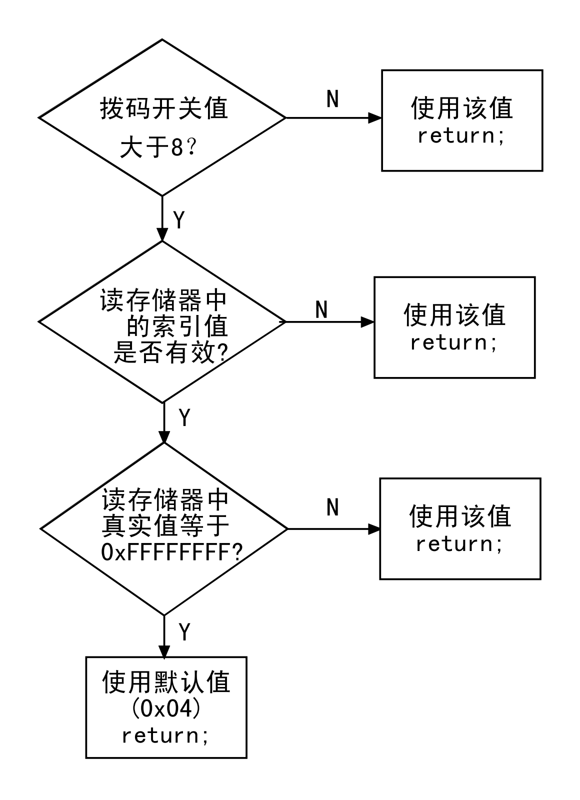


图 6　波特率设置顺序

其中设置节点ID的拨码开关使用7位，当其值为0时，表示拨码开关无效，取值范围为1~127之间。波特率索引值使用4位拨码，当该拨码开关值大于8时无效，其有效的取值范围为0~8，波特率索引值与标准波特率的对应关系如表 5所示。如果该表格中的标准波特率无法满足需求，则用户可以自己计算波特率值（32位长度的波特率定时参数）来设置模块，当拨码开关和波特率索引值都无效时，则模块会自动检测该值是否有效（值 为0xFFFFFFFF和0x00000000时无效），如果已经设定波特率索引值则优先使用该值初始CAN通信波特率。其中CAN控制器的工作频率为24MHz。

表 5　波特率索引与标准波特率关系

波特率索引值	标准波特率
0	1Mbps
1	800Kbps
2	500Kbps
3	250Kbps
4	125Kbps
5	100Kbps
6	50Kbps
7	20Kbps
8	10Kbps

3.2　通过层设置配置节点地址和波特率（LSS DS305）

在CiA DSP-305标准中，CANopen定义了LSS（Layer Setting Services），允许在不提供任何方式的外部机械配置（例如，通过DIP或HEX开关）的设备配置基本参数（波特率、节点编号）。LSS主机可以在CAN总线上改变一个CANopen LSS从机的波特率和节点编号。首先，LSS主机使所有LSS从机进入配置模式。然后，LSS主机使用‘配置位时序（Configure Bit Timing）’服务发送新的波特率。这时，LSS从机以一个指示它是否支持这个新波特率的CAN报文来响应。如果该LSS从机接受这个新的波特率，LSS主机就给LSS从机发送‘启动位时序（Activate Bit Timing）’服务。在一个称为‘switch_delay’的时间过后，通知LSS从机激活这个新的波特率。这个周期成功结束后，LSS主机让LSS从机回到工作模式。

LSS服务也可以用来改变一个LSS从机的节点地址。为了到达目的，LSS主机让所有的LSS从机再次进入配置模式。然后，LSS主机发送新的节点地址。这时，LSS从机以一个指示这个新节点编号是否在这个节点所能支持的节点编号范围之内的CAN报文来响应。在LSS从机切换回到工作模式时，发出一个软件复位。这样来使LSS从机根据新的节点编号来配置它的通信对象。

1.　配置节点地址和波特率参数

如表 6、表 7、表 8、表 9、表 10和表 11所示，为配置当前节点参数的命令，注意在启用LSS之前必须先切换到配置模式。如果没有保存当前的参数则掉电之后会丢失当前配置参数，保存命令如表 13和表 14所示。

表 6　LSS 配置模式切换

标识符 (COB-ID)	DLC	数据							
		0	1	2	3	4	5	6	7
0x7E5	8	0x04	mod	保留					

mod：新的LSS模式

0 ＝ 切换到工作模式

1 ＝ 切换到配置模式

表 7 配置波特率

标识符 (COB-ID)	DLC	数据							
		0	1	2	3	4	5	6	7
0x7E5	8	0x13	tab	ind	保留				

tab：指示要使用的波特率表

0 ＝ 根据CiA DSP-305定义的波特率表

1...127 ＝ 保留

128...255 ＝ 可以由用户来定义

ind：波特率表内的索引，这个波特率表保存着CANopen设备的新波特率

表 8 配置波特率响应

标识符 (COB-ID)	DLC	数据							
		0	1	2	3	4	5	6	7
0x7E4	8	0x13	err	spec	保留				

err：错误码

0 ＝ 操作成功结束

1 ＝ 不支持波特率

2..254 ＝ 保留

255 ＝ spec中的特定错误码

spec：生产商指定的错误码（只有在err＝255时）

表 9 激活新设置波特率

标识符 (COB-ID)	DLC	数据							
		0	1	2	3	4	5	6	7
0x7E5	8	0x15	delay		保留				

delay：激活新波特率之前相对时间（以ms为单位）

表 10 配置节点地址

标识符 (COB-ID)	DLC	数据							
		0	1	2	3	4	5	6	7
0x7E5	8	0x11	nid	保留					

nid：LSS从机的新节点地址（允许的值：1～127）

表 11 配置节点地址响应

标识符 (COB-ID)	DLC	数据							
		0	1	2	3	4	5	6	7
0x7E4	8	0x11	err	spec	保留				

err：错误码

0 ＝ 操作成功结束

1 ＝ 节点地址无效（允许的值：1～127）

2...254 ＝ 保留

255 ＝ spec中的特定错误码

spec：生产商指定的错误码（只有在err＝255时）

表 12　根据 CiA DSP-305 得到的波特率与索引的关系

表索引	波特率
0	1000 kBit/s
1	800 kBit/s
2	500 kBit/s
3	250 kBit/s
4	125 kBit/s
5	100 kBit/s
6	50 kBit/s
7	20 kBit/s
8	10 kBit/s

2.　保存LSS设置参数

通常情况下，在使用了DIP开关或其它的HEX开关来设置当前的节点地址和波特率，则优先使用这些值，LSS设置的参数在重新上电或复位之后无法被启用，即LSS设置只在当前通信中有效。

在LSS设置完成当前节点的ID和波特率之后如果需要在之后的使用过程继续使用这组值（仅在DIP开关设置的值无效情况下），则需要把参数存储在存储器中，其LSS操作命令如表13和表 14所示。

表 13　LSS 保存参数

标识符 (COB-ID)	DLC	数据							
		0	1	2	3	4	5	6	7
0x7E5	8	0x17	保留						

表 14　LSS 保存参数响应

标识符 (COB-ID)	DLC	数据							
		0	1	2	3	4	5	6	7
0x7E5	8	Erro code	Spec .error	保留					

Erro code:

0：执行成功

1：保存参数不支持

2：存储介质出现错误

3~254：保留

255：执行过程中出现错误

Spec. error:

255：执行过程中的错误代码

其它：保留

4. XGate-COP10 中使用CANopen协议（DS301）

4.1 XGate-COP10 预定义连接

在XGate-COP10中使用了对象字典的0x1000~0x1FFF和厂商自定义区0x2000~0x5FFF的区域。这些对象字典负责着CANopen与CAN网络上的其它应用数据的通信和数据交换，对象字典使用了索引与子索引来定义，每个对象字典项都有自己的数据长度（UIN8，UINT16，UIN32等）和属性（RO、WO、RW、CONST、MAPPALE）。这些对象字典的数据可以通过SDO服务来修改，当然也只有这些项的属性必须是WO或RW才能修改。

如表 15所示为XGate-COP10所使用的部分对象字典及其属性，完整的对象字典请参考附录A。

通常情况下，在一个典型的CANopen网络中，有一个CANopen主站和若干个CANopen从站，这种情况下通常使用CANopen预定义连接。所谓的预定义连接是指与通信相关的COB-ID与节点ID相关联。

例：假设当前节点ID为0x20，则TPDO10的COB-ID为0x180+0x20(0x180+NodeID)。具体的预定义连接集如表 16所示。

表 15　部分对象字典定义

索引	子索引	名称	类型	属性	是否可映射	存储	单位
0x1000	0x00	DeviceType	UINT32	RO	NO	NO	—
0x1008	0x00	DeviceName	String	RO	NO	NO	—
0x100C	0x00	Guard Time	UINT16	RW	NO	YES	Ms
0x1400	0x00	Subindex num	UINT8	RO	NO	NO	—
	0x01	COB-ID	UINT32	RW	NO	YES	—
	0x02	Trans type	UINT8U	RW	NO	YES	—
……							
0x2000	0x00	Subindex num	UINT8	RO	NO	NO	—
	0x01	Input buffer #0	UINT8	RW	YES	NO	—
	……	……	……	……	……	……	……

注意:有些对象字典的参数可以保储在存储器中，对于对象字典 0x1000~0x1FFF中的参数，如果选择使用预定义连接方式则存储的参数无效（节点地址改变之后，存储的参数无效，节点将使用预定义连接集），这些参数将使用预定义连接集定义的值。预定义连接集定义如表 16所示。

表 16　预定义的主机/从机连接集

对象	功能码	节点地址	COB-ID	对象字典索引
广播报文				
NMT	0000	-	0	-
SYNC	0001	-	0x80	0x1005, 0x1006, 0x1007
TIME STAMP	0010	-	0x100	0x1012, 0x1013
点对点报文				
紧急报文	0001	1-127	0x81-0xFF	0x1014, 0x1015
TPDO1	0011	1-127	0x181-0x1FF	0x1800
RPDO1	0100	1-127	0x201-0x27F	0x1400

TPDO2	0101	1-127	0x281-0x2FF	0x1801
RPDO2	0110	1-127	0x301-0x37F	0x1401
TPDO3	0111	1-127	0x381-0x3FF	0x1802
RPDO3	1000	1-127	0x401-0x47F	0x1402
TPDO4	1001	1-127	0x481-0x4FF	0x1803
RPDO4	1010	1-127	0x501-0x57F	0x1403
默认 SDO (tx)	1011	1-127	0x581-0x5FF	0x1200
默认 SDO (rx)	1100	1-127	0x601-0x67F	0x1200
NMT 错误控制	1110	1-127	0x701-0x77F	0x1016, 0x1017

4.2 XGate-COP10 操作

4.2.1 网络管理服务（NMT）

1. 网络控制（NMT Module Control）

XGate-COP10支持DS301所定义的网络管理命令，这些网络管理命令可以是CANopen主站发出也可以是其它的从节点发出。其操作命令如下表 17所示，其中当Node_ID=0时，则所有的从站设备被控制(广播方式)，CS为命令字对应着不同的控制动作如表 18所示。

表 17 NMT 控制合令

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1
0x000	2	CS(命令字)	NodeID(节点号)

表 18 NMT 命令字及相应功能服务

CS（命令字）	NMT 服务（控制动作）
0x01	启动从站设备
0x02	停止从节点设备
0x80	使从站进入预操作
0x81	复位从节点
0x82	复位节点通信

例：需要启动CANopen网络中所有的节点，可使用如下表 19所示的命令。

表 19 NMT 启动从节点

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1
0x000	2	0x01	0x00

如果需要控制网络中某个具体设备，使其进入到预操作状态，假设节点地址为0x20，则命令如表 20所示。

表 20 NMT 启动从节点

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1
0x000	2	0x80	0x20

2. 节点保护（NMT Node Guarding）

通过节点保护服务，NMT主节点可以检查每个节点的当前状态，当这些节点没有数据传送时这种服务尤其有意义。主节点通过发送远程帧来触发相应从节点的节点保护，其命令格式如表 21所示。

主节点→从节点(命令)：

表 21 NMT 主节点保护命令帧(远程帧)

COB-ID(CAN-ID)	DLC
0x700 + NodeID	1

从节点→主节点(响应)：

表 22 NMT 从节点应答帧

COB-ID(CAN-ID)	DLC	BYTE0
0x700 + NodeID	1	Bit7:触发位，Bit0~Bit6 状态

其中Byte0中的最高位（bit7）为触发位，即从站每发一帧应答就会交替变化（0、1），以示帧与帧之间的区别，其中Bit0~Bit6为从节点的状态，该值所表达从站所处的状态如表 23所示。

表 23 节点保护状态值

值（Value）	所处状态
0x00	初始化（Initialising）
0x04	停止状态（Stopped）
0x05	操作状态（Operational）
0x7F	预操作状态（Pre-operational）

例：假设主节点需要对节点号为0x20从节点进行节点保护，其命令如表 24和表 25所示。

主节点→从节点：

表 24 保护节点(远程帧)

COB-ID(CAN-ID)	DLC
0x720	1

从节点→主节点：

表 25 从节点(0x20)应答帧

COB-ID(CAN-ID)	DLC	BYTE0
0x720	1	0x85

其中BYTE0的Bit7 = 1，状态=0x05，表示节点号为0x20的从站正处于操作状态。

3. 寿命保护（NMT Life Gartding）

其中节点保护主要针对的是NMT主节点获取从节点的状态，而寿命保护主要是节点对另一节点的监控。寿命保护包括两个参数，即保护时间和生命因子，启用寿命保护的节点接收来自另一节点的远程帧（远程帧格式与节点保护帧格式相同如表 21），启用寿命保护的节点接收到该远程帧则应答该节点的状态（应答帧格式如表 22所示）。

寿命保护的两个参数：保护时间和生命因子（分别位于对象字典的0x100C和0x100D）构成了节点的寿命时间（即寿命时间=保护时间x 生命因子），保护时间的单位为毫秒，如果两个参数中有一个为0则表示寿命保护未启用。如果在保护时间内未接收到远程帧则会出现”Message Lost”的提示信息，在寿命时间内未接收到远程帧则会出现” Connection Lost”信息，这些信息均在调试串口中打印出来，同时错误指示灯出现“闪烁两下”以示当前的寿命保护丢失。

4. 启动报文(NMT Boot-up)

当XGate-COP10初始化完成(Boot-up)，就会发送一个标识报文，其报文格式如表 26所示。

表 26 初始化完成标识报文格式

COB-ID(CAN-ID)	DLC	BYTE0
0x700 +NodeID	1	0x00

例：假设XGate-COP10的节点号为0x20，则发送的启动文如表 27所示。

表 27 初始化完成标识报文

COB-ID(CAN-ID)	DLC	BYTE0
0x720	1	0x00

5. 心跳报文（Heartbeat Producer）

心跳报文分为生产者和消费者，在XGate-COP10模块中只支持心跳报文生产，即XGate-COP10可以生产心跳报文。该参数在对象字典0x1017中定义（数据长度16位，单位：毫秒），其心跳报文如表 22所示，与节点保护和寿命保护的应答帧相同。

例：假设节点地址为0x20，处于操作状态，0x1017中的参数设置为100，则该从节点每隔100毫秒发送一帧如表 28所示的报文。

表 28 从节点(0x20)心跳报文

COB-ID(CAN-ID)	DLC	BYTE0
0x720	1	0x05

注意：在同一个 XGate-COP10 模块中同一时间寿命保护和心跳报文不能同时使用。

4.2.2 同步报文对象(SYNC)

同步报文分为消费和生产，在XGate-COP10中只支持同步报文的消费，即接收来自主节点或其它节点的同步报文，同步报文的帧结构如表 29所示。对象字典的0x1005定义了接收同步报文的COB-ID，在CANopen预定义连接集里定义其值为0x80，对象字典的0x1007定义了同步的时间窗口（在接收到同步报文后要求更新相应数据的最长时间间隔）。同步报文主要应用在PDO接收和发送的过程中，其使用方法在以下PDO数据发送和接收过程中详细介绍。

表 29 同步报文帧格式（远程帧）

COB-ID(CAN-ID)	DLC
0x80	0

4.2.3 紧急报文对象(EMCY)

在XGate-COP10中支持紧急报文，即在XGate-COP10内部出现错误时发送该报文，其报文格式如表 30所示。其中紧急错误码指定当前出现的错误的具体类型。错误寄存器存放当前错误类型，根据该值可以判断出当前紧急报文所代表的错误类型，其值定义如表 31所示。

表 30　紧急报文帧格式

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1	BYTE2	BYTE3 －BYTE7
0x80 + NodeID	8	紧急错误码		错误寄存器	生产厂商指定的信息
		索引 0x1003		索引 0x1001	

表 31　错误寄存器

位(Bit)	错误类型
0	普通错误(Generic)
1	电流错误(Current)
2	电压错误(Voltage)
3	温度错误(Temperature)
4	通信错误(Communication)
5	设备描述错误(Device profile specific)
6	Reserved(=0)
7	生产厂商定义的错误(Manufacturer specific)

紧急错误代码含义如表 32所示。

表 32　紧急错误代码

应急错误代码	代码功能描述
00xx	Error Reset 或 No Error
10xx	Generic Error
20xx	Current
21xx	Current，device input side
22xx	Current，inside the device
23xx	Current，device output side
30xx	Voltage
31xx	Mains voltage
32xx	Voltage inside the device
33xx	Output voltage
40xx	Temperature
41xx	Ambient temperature
42xx	Device emperature
50xx	Device hardware
60xx	Device software
61xx	Internal software
62xx	User software

63xx	Data set
70xx	Additional modules
80xx	Monitoring
81xx	communication
8110	CAN overrun
8120	Error Passive
8130	Life Guard Error 或Heartbeat Error
8140	Recovered from Bus-Off
82xx	Protocol Error
8210	PDO no processed Due to length error
8220	Length exceedd
90xx	External error
F0xx	Additional functions
FFxx	Device specific

例：假设节点地地址为0x20，CAN总线错误超过警戒值则出现" Error Passive"的警告，则XGate-COP10发送紧急报文如表 33所示。

表 33　紧急报文（总线错误）

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1	BYTE2	BYTE3 －BYTE7
0xA0	8	0x20	0x81	0x11	0x00000000

注意：XGate-COP10 模块发生紧急情况，这些错误主动发出。

4.2.4 服务数据对象(SDO)

对象字典充当应用层和通信层之间的主要数据交换媒介。一个CANopen设备的所有数据项可以在对象字典中被管理。每个对象字典项可以使用索引和子索引来定位。CANopen定义通常所说的服务数据对象（SDO）来访问这些项。

XGate-COP10支持1个SDO服务器，即可以提供SDO服务，且SDO使用预定义连接的发送和接收COB-ID，0x580 + NodeID(发送)和0x600 + NodeID(接收)。SDO分为加速传输、段传输、和块传输。因为在XGate-COP10中SDO的加速传输经常会使用，所以本说明文档重点介绍加速传输，其它的传输类型可查阅CANopen　DS301及相关的协议文档。

1.　SDO数据传输

加速传输一帧最多只能传输4个字节的数据，报文基本结构如表 34和表 35所示，通过SDO的命令字来区分该帧数据的类型。

表 34　SDO 报文格式（客户端→服务器）

COB-ID(CAN-ID)	DLC	Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x600 + NodeID	8	CMD(SDO 命令字)	对象索引	对象子索引	**

表 35 SDO 应答格式（服务器→客户端）

COB-ID(CAN-ID)	DLC	Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x580 + NodeID	8	CMD(SDO 命令字)	对象索引	对象子索引	**

下载（Download）是指客户端对服务器（从站）对象字典进行写操作，上传（Upload）指客户端对服务器对象字典进行读操作。

表 36 启动域下载（Initiate Domain Download）

CMD(SDO 命令字)								
Bit	7	6	5	4	3	2	1	0
Client→ Server	0	0	1	-	n		e	s
Client←Server	0	1	1	-	-	-	-	-

说明：其中“-”项为不相关项，通常设置为0

- n ： 如果 e=1，且 s=1，则有效，否则为 0；表示数据部分中无意义数据的字节数（字节 [8 – n] 到 7 数据无意义）；
- e ： 0 = 正常传送，1 = 加速传送；
- s ： 是否指明数据长度，0 = 数据长度未指明，1 = 数据长度指明；
- e = 0， s = 0： 由 CiA 保留；
- e = 0， s = 1 ： 数据字节为字节计数器，byte 4 是数据低位部分（LSB），byte 7 是数据高位部分（MSB）；
- e = 1 ： 数据字节为将要下载（download）的数据。

表 37 启动域上传 Initiate Domain Download）

CMD(SDO 命令字节)								
Bit	7	6	5	4	3	2	1	0
Client→ Server	0	1	-	-	-	-	-	-
Server→ Client	0	1	1	-	n		e	s

说明：命令字与表 36相同。

例：假设当前XGate-COP10的地址为0x20，对对象字典进行0x1800 03进行读写操作。

- 向 0x1800 03 写入 0x3E8，操作过程如表 38和表 39所示。

表 38 向对象字典 0x1800 03 写入 0x3E8 命令

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte4	Byte 5	Byte 6	Byte 7
0x620	8	0x2B	0x00	0x18	0x03	0xE8	0x03	0x00	0x00

表 39 向对象字典 0x1800 03 写入 0x3E8 后的应答

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte4	Byte 5	Byte 6	Byte 7
0x5A0	8	0x60	0x00	0x18	0x03	0x00	0x00	0x00	0x00

● 读对象字典 0x1800 03 中的数据，其命令与应答如表 40和表 41所示。

表 40 读取对象字典 0x1800 03 数据命令

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte4	Byte 5	Byte 6	Byte 7
0x620	8	0x40	0x00	0x18	0x03	0x00	0x00	0x00	0x00

表 41 读对象字典 0x1800 03 后的应答帧

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte4	Byte 5	Byte 6	Byte 7
0x5A0	8	0x4B	0x00	0x18	0x03	0xE8	0x03	0x00	0x00

2. SDO中止服务

在SDO的传输过程中当出现错误，SDO的客户端和服务器都可以发送这个报文来通知对方中止当前的操作，中止报文的格式如表 42和表 43所示。

表 42 中止报文格式

COB-ID(CAN-ID)	DLC	Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x600 + NodeID/0x580+NodeID	8	CMD(SDO 命令字)	索引	子索引	**

表 43 中止报文命令字定义

CMD(SDO 命令字节)								
Bit	7	6	5	4	3	2	1	0
Client→ Server/ Client→ Server	1	0	0	-	-	-	-	-

例：假设当前节点的NodeID为 0x20，在读取数据过程中出现错误中止，则会返回相应的错误代码，如下表 44和表 45所示。

表 44 读取不存在的对象字典 0x6000 00

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte4	Byte 5	Byte 6	Byte 7
0x620	8	0x40	0x00	0x60	0x00	0x00	0x00	0x00	0x00

表 45 读取不存在的对象字典 0x6000 00 返回数据

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte4	Byte 5	Byte 6	Byte 7
0x5A0	8	0x80	0x00	0x60	0x00	0x00	0x00	0x02	0x06

注意：返回数据为一中止传送的错误代码，SDO错误代码为 32 位长度的数据，该帧数据应答的错误代码为 0x06020000，代表的含义为“对象字典不存在”。具体的中止错误代码含义可参照表 46。

表 46　中止错误代码表

中止代码	代码功能描述
0503 0000	触发位没有交替改变
0504 0000	SDO 协议超时
0504 0001	非法或未知的 Client/Server 命令字
0504 0002	无效的块大小（仅 Block Transfer 模式）
0504 0003	无效的序号（仅 Block Transfer 模式）
0503 0004	CRC 错误（仅 Block Transfer 模式）
0503 0005	内存溢出
0601 0000	对象不支持访问
0601 0001	试图读只写对象
0601 0002	试图写只读对象
0602 0000	对象字典中对象不存在
0604 0041	对象不能够映射到 PDO
0604 0042	映射的对象的数目和长度超出 PDO 长度
0604 0043	一般性参数不兼容
0604 0047	一般性设备内部不兼容
0606 0000	硬件错误导致对象访问失败
0606 0010	数据类型不匹配，服务参数长度不匹配
0606 0012	数据类型不匹配，服务参数长度太大
0606 0013	数据类型不匹配，服务参数长度太短
0609 0011	子索引不存在
0609 0030	超出参数的值范围(写访问时)
0609 0031	写入参数数值太大
0609 0032	写入参数数值太小
0609 0036	最大值小于最小值
0800 0000	一般性错误
0800 0020	数据不能传送或保存到应用
0800 0021	由于本地控制导致数据不能传送或保存到应用
0800 0022	由于当前设备状态导致数据不能传送或保存到应用
0800 0023	对象字典动态产生错误或对象字典不存在（例如，通过文件生成对象字典，但由于文件损坏导致错误产生）

4.2.5 过程数据对象（PDO）

过程数据对象（PDO）用作传输实时数据，传输模型采用生产者—消费者模型如图 2所示，PDO的接收者可以是主节点也可以是其它的从节点，且不需要应答。

在XGate-COP10中最多可支持12个TPDO（索引范围从0x1800~0x180B）和12个RPDO(0x1400~0x140B)，在出厂时预定义连接集所定义的4个TPDO(索引范围0x1800~0x1803)和4个RPDO(0x1400~0x1403)可用。

1. 过程数据接收（RPDO）

在XGate-COP10出厂时已经为每个PDO预定义了映射对象，如图 7所示，全部RPDO的数据映射项都默认连接到XGate-COP10 8bit输出区。即当RPDO接收到来自网络的数据之后，把数据更新到所对应的输出数据区，当数据更新完成之后，XGate-COP10会给出一个中断信号（高电平->低电平），当数据未被读取时中断信号引脚将一址保持低电平，数据被读出之后中断引脚将保持在高电平。

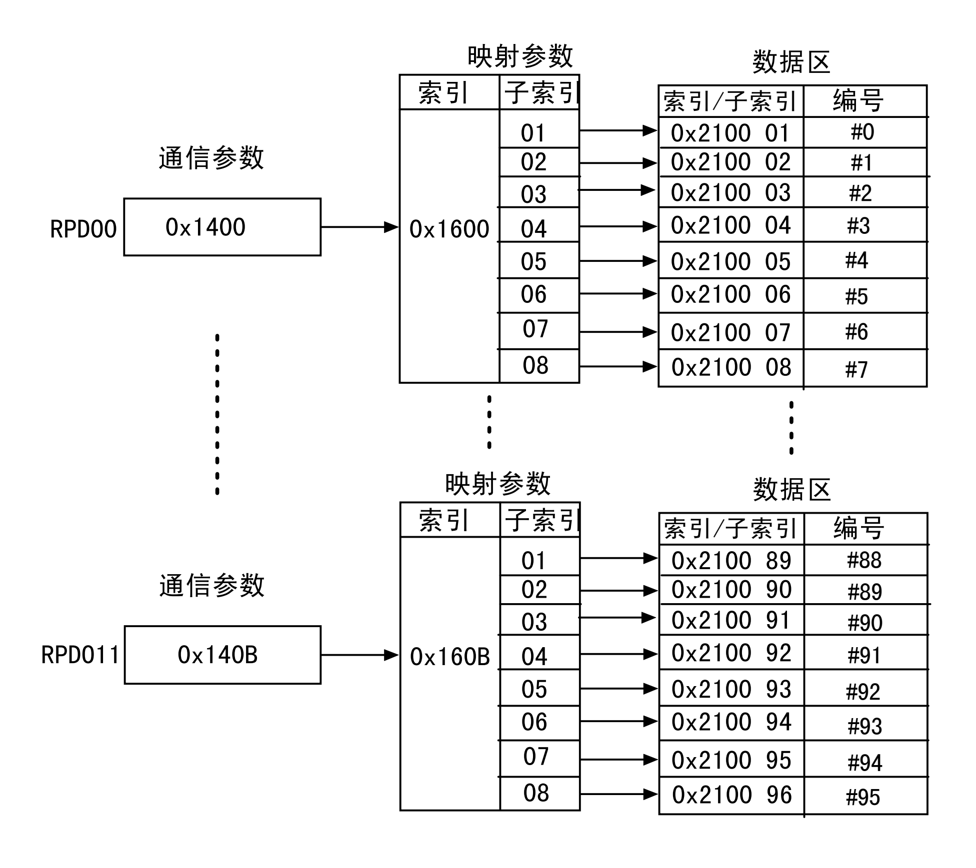


图 7 RPDO 映射关系

- 通信参数

RPDO的通信参数相比TPDO来说比较少，RPDO只有传输类型（transmission type）一项对应通信。其值定义如表 47所示。

表 47 RPDO 的传输类型

传输类型	PDO 接收	数据更新
0	PDO 将一直被接收，分析。如果需要，在接收到下个有效的 SYNC 报文时对数据进行更新。	在接收到一个 SYNC 报文时对数据进行分析。如果与之前的 RPDO 相比数据已经更改了，那么数据将在输出上被更新。SYNC 报文的传输是非循环的。
1－240		在接收到第 n 个编号的 SYNC 报文时对数据进行分析。如果与之前的 RPDO 相比数据已经更改了，那么数据将在输出上被更新。传输类型与值 n 相对应。SYNC 报文的传输是循环的。
241－251	保留	
252	保留	

253		
254	PDO 将一直被接收到。	应用定义更新输出数据的事件。
255	PDO 将一直被接收到。	设备子协议定义更新输出数据的事件。

例：假设XGate-COP10 节点为 0x20，采用预定义连接，则RPDO0 的COB-ID为 0x220。则其接收的TPDO COB-D也应为 0x220 如表 48所示，该TPDO正好与节点NodeID为 0x20 的RPDO1 的COB-ID相同，则该RPDO接收这帧PDO数据，并且按照图 7所示的映射关系图把数据更新到数据输出区，最后输出缓冲区对应的数据如表 49所示。

表 48　RPDO0 接收其它节点的 TPDO

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte4	Byte 5	Byte 6	Byte 7
0x220	8	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88

表 49　数据区数据

数据区编号	#0	#1	#2	#3	#4	#5	#6	#7
数据	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88

2.　过程数据发送（TPDO）

在XGate-COP10 中最多支持 12 个TPDO，在出厂时预定义的 4 个PDO可用，即TPDO0~TPDO3，其中TPDO4~TPDO11 不可用。预定义的TPDO都已经在出厂时已经预定义了映射参数，分别映射到数据输入区 0x2000 01 ~0x2000 96，如图 8所示。

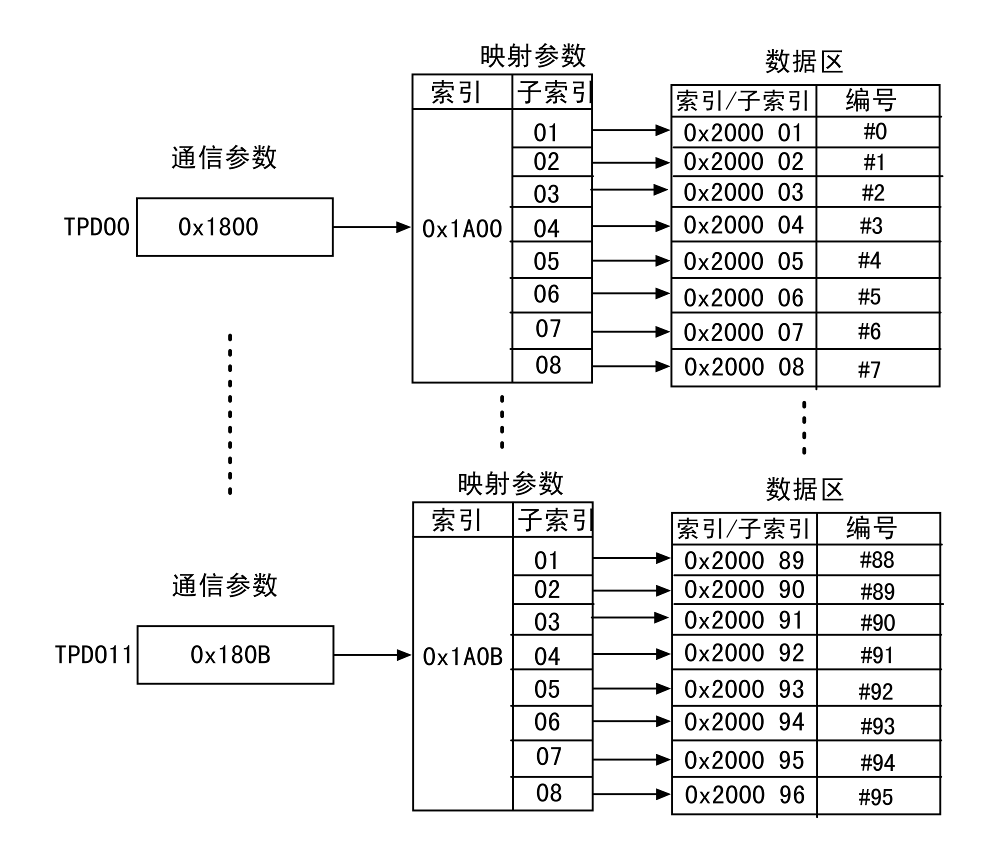


图 8　TPDO 映射关系

- 通信参数

每个TPDO都包含有相应的通信参数，这些通信参数决定着TPDO发送的类型以及发送的触发条件等。其中参数主要包含有三种，分别为传输类型、禁止时间以及事件定时。

➢ 传输类型（Transmission type）

传输类型定义了该TPDO传输方式，通信参数的子索引2定义该对象，具体值定义如表 50所示。

表 50　TPDO 传输类型

传输类型	数据请求	发送 PDO
0	数据（输入值）在接收到一个 SYNC 报文时被读取。	如果与之前的 PDO 内容相比 PDO 数据已经更改了，那么 PDO 将被发送。
1－240	数据接收第 n 个编号的 SYNC 报文时被收集和更新，然后在总线上发送。传输类型对应值 n。	
241－251	保留	
252	数据（输入值）在接收到一个 SYNC 报文时被读取。	PDO 在请求时通过一个远程帧被发送。
253	应用持续收集和更新输入数据。	
254	应用定义引发数据请求和 PDO 传输的事件。造成 PDO 传输的事件可以是事件定时器的时间已到。事件定时器周期用子索引 5 来配置。PDO 传输（与事件和事件定时器是否被配置都无关）总是启动一个新的事件定时器周期。	
255	设备子协议定义引发数据请求和 PDO 传输的事件。造成 PDO 传输的事件可以是事件定时器的时间已到。事件定时器周期用子索引 5 来配置。PDO 传输（与事件和事件定时器是否被配置都无关）总是启动一个新的事件定时器周期。	

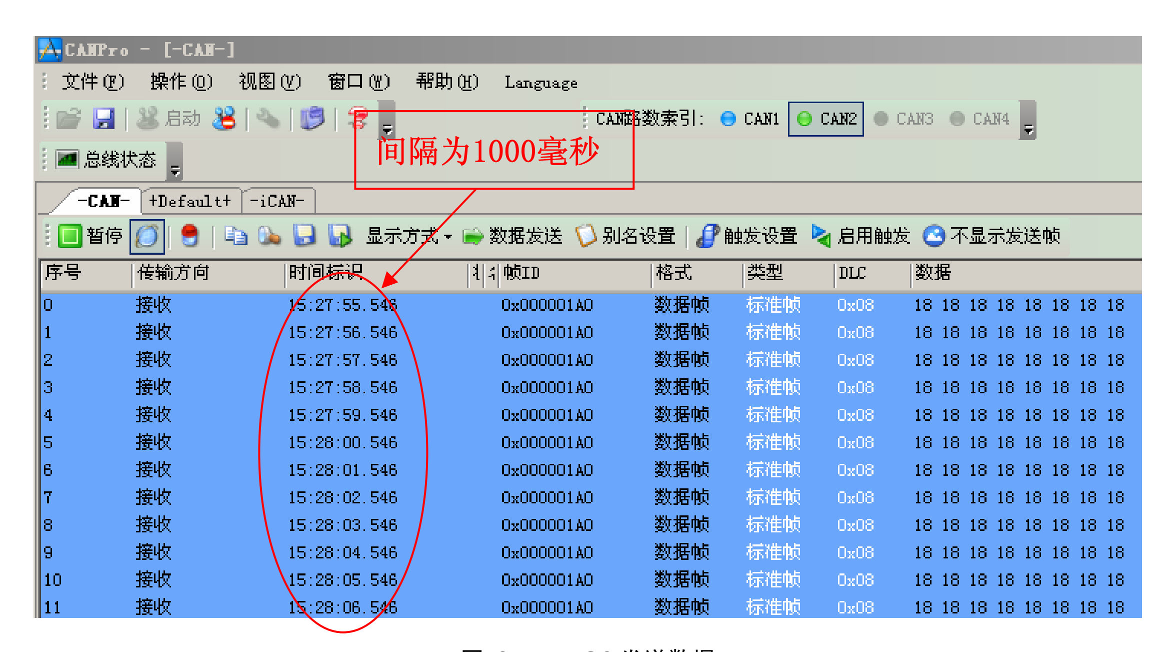
➢ 禁止时间(Inhibit Time)

禁止时间的定义是为了防止TPDO发送过于频繁而占用大量的总线带宽，从而影响到总线通信。因而定义了同一个TPDO发送PDO的最短时间间隔（单位为毫秒），当该参数为0时无效，在通信参数子索引3中定义。

➢ 定时时间(Event Time)

定时时间参数定义了该PDO的发送循环时间（单位为毫秒），需要PDO的传输类型设置为254或255，当该参数为0时无效，在通信参数子索引5中定义。

例：假设当前的节点NodeID为0x20，TPDO0定时时间（Event time）参数为1000，传输类型（Transmission type）为254，数据输入区#0~#7号的数据为0x18，则TPDO0发送数据如图 9所示。



序号	传输方向	时间标识	帧ID	格式	类型	DLC	数据
0	接收	15:27:55.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
1	接收	15:27:56.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
2	接收	15:27:57.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
3	接收	15:27:58.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
4	接收	15:27:59.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
5	接收	15:28:00.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
6	接收	15:28:01.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
7	接收	15:28:02.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
8	接收	15:28:03.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
9	接收	15:28:04.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
10	接收	15:28:05.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18
11	接收	15:28:06.546	0x000001A0	数据帧	标准帧	0x08	18 18 18 18 18 18 18 18

图 9　TPDO0 发送数据

5. 输入输出数据缓冲区

XGate-COP10拥有96字节的输入输出缓冲区。为了使用户操作方便，对输入输出区定义了双字节和四字节的输入输出数据区，对应的单字节(8bit)、双字节(16bit)和四字节(32bit)的数据区都占用相同的内存空间，所有数据均采用小端数据格式。

表 51所示。在XGate-COP10出厂时已经默认把8bit区数据映射到相应的PDO中，如果用户需要使用16bit或32bit的数据区，则可更改相应的映射参数。

例：8bit区的数据编号#0(0x11)，#1(0x22)，#2(0x33)，#3(0x44)，则对应的16bi区编号为＃0(0x2211)，#1(0x4433)，32bit区编号为#0(0x44332211)。

注意：加"#"号的数据为数据缓冲区的编号，括号内为该缓冲区中的数据。

表 51　输入输出数据区对应关系

数据输入区													
8bit	#0(0x12)	#1(0x34)	#2(0x56)	#3(0x78)	#4(0x89)	#5(0x10)	#6(0x23)	#7(0x45)	…	#92	#93	#94	#95
16bit	#0(0x3412)		#1(0x7856)		#2(0x1089)		#3(0x4523)		…	#46		#47	
32bit	#0(0x78563412)				#1(0x45231089)				…	#23			
数据输出区													
8bit	#0(0x12)	#1(0x34)	#2(0x56)	#3(0x78)	#4(0x89)	#5(0x10)	#6(0x23)	#7(0x45)	…	#92	#93	#94	#95
16bit	#0(0x3412)		#1(0x7856)		#2(0x1089)		#3(0x4523)		…	#46		#47	
32bit	#0(0x78563412)				#1(0x45231089)								

6. XGate-COP10 串口操作

6.1　串口通信协议

XGate-COP10与用户通信采用异步串口进行通信，通信模式为半双工，通信信号为TTL电平，通信协议采用广州致远电子有限公司自定义串行通信协议。

6.1.1 异步串口数据帧格式

每1 个字节用10 bits 传送，1 个启始位、8 个数据位、无奇偶校验位、1 个停止位，因定波特率为115200 bps。

应答方式：用户设备主动询问（主），XGate-COP10被动回答（从）。

主/从应答帧结构实现通常的数据通讯，数据的通讯由主机发起，称为命令帧；从机接收到后进行应答，称为响应帧。

表 52 命令帧格式

1Byte	1Byte	1Byte	1Byte	nByte	1Byte
起始字	命令码	命令信息(长度)	特定参数	命令数据	校验码
0x7E	CMD	CMDinfo	SpeByte	DATA	CRC

表 53 应答帧格式

1Byte	1Byte	1Byte	1Byte	nByte	1Byte
起始字	响应码	响应信息(长度)	特定参数	响应数据	校验码
0x7E	ACK	ACKinfo	SpeByte	DATA	CRC

下面按各命令排序，详细介绍命令帧、响应帧的规则。

命令帧、应答帧的总长度为：命令/响应信息的CMDinfo/ ACKinfo （数据长度）+ 5Byte，各字段说明如下：

- 帧起始字符 SOF，固定为 0x7E，长度为一个字节。
- 命令 CMD/响应码 ACK，通常 CMD=ACK，长度为一个字节。
- 命令信息 CMDinfo/响应信息 ACKinfo 指出命令信息/响应信息的长度(字节)，不包括本身。CMDinfo/ACKinfo = 0 表示没有数据，CMDinfo/ACKinfo = 1 表示本帧含有 1 个字节数据。
- 特殊参数SpeByte包括Error 、AllDataSegSize、DataSegNum信息，具体含义如表 54

Error为特定意义位。在命令信息中，Error为保留位，通常Error =0；在响应帧中，Error为错误标识位，Error =1，表明命令执行出错，DATA区跟随错误代码，Error =0，表明请求成功，DATA区跟随应答数据。

AllDataSegSize、DataSegNum指出分段信息，当数据量超过255字节时，需要使用多帧来传送，此时使用分段传输方式，共可以分为7段，即7帧。

表 54 特定参数（SpeByte） 定义

BIT.7	BIT.6	BIT.5	BIT.4	BIT.3	BIT.2	BIT.1	BIT.0
Error	AllDataSegSize			保留位	DataSegNum		

AllDataSegSize表示本次信息数据共有几帧；DataSegNum表示本次发送的是第几帧该信息数据。当AllDataSegSize = DataSegNum时表示该信息数据传输完毕。AllDataSegSiz和

DataSegNum最小值为1，不能为0。

本协议将不使用分段，AllDataSegSize、DataSegNum 固定为“1”。

- 命令/响应数据 DATA，此部分与命令/响应码相结合，描述数据的具体含义。长度在 CMDinfo/ ACKinfo 中说明，最大为 255 个字节/帧（仅数据区字节数）。
- 校验和 CRC：命令/响应数据的校验和，长度为一个字节。校验和为前面所有数据的异或值。CRC 的计算公式如下：

CRC ＝ 0x7E^ CMD ^ CMDinfo ^ SpeByte ^DATA[0]^DATA[1]^…^DATA[n-1] 或

CRC ＝ 0x7E^ ACK ^ ACKinfo ^ SpeByte ^DATA[0]^DATA[1]^…^DATA[n-1]

6.2 XGate-COP10 操作命令

用户通过XGate-COP10的通信串口对设备进行操作，操作命令如6.2.1节所示，本节所有的操作命令都认为成功执行，并得到正确执行的命。如果出现错误，其错误响应帧和错误代码将在6.3节中介绍。

6.2.1 读取设备信息(命令码：0x01)

通过串口用户可以读取用户设置的设备信息，该设备信息是用户设置的信息，位于对象字典的0x2404。其中设备类型位于对象字典0x1000 00，因为该设备是通用设备，没有使用标准设备描述，因此该参数按照CiA定义应该为0x00000000,建议用户不要更改此参数。操作命令及响应帧如表 55和 表 56所示，n表示该命令帧数据长度。其中命令数据域的第一字节为操作模式，根据该字节判断该命令所读取的信息类别，以下将详细介绍操作命令。

注意：通过串口读取的设备信息并不是 XGate-COP10 设备信息，而是用户设备相关的设备信息。

表 55 读取设备信息命令

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x01	n	0x11	DAT	校验码

表 56 读取设备信息响应

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x01	n	0x11/0x91	DAT	校验码

1. 读取设备类型(操作模式：0x01)

设备类型位于对象字典的0x1000 00，数据长度4字节，高2字节为XGate-COP10设备类型，低2字节为用户设备类型代码（用户设置）。

例：如下命令读回设备类型为0x0A000011，即当前的的XGate-COP10的设备类型为0x00A0，用户的设备类型为0x0000。

命令：0x7E 0x01 0x01 0x11 **0x01** 0x6E

响应：0x7E 0x01 0x05 0x11 **0x01 0x11 0x00 0x0A 0x00** 0x71

注意：使用该模块建议不要更改设备类型，因为XGate-COP10没有使用标准设备描述，因此设备类型为通常为0x00000000。

2. 读硬件版本(操作模式:0x02)

硬件版本信息位于对象字典0x2404 01，数据长度为4字节，该值为用户设置信息。

例：如下命令读取回来的硬件版本信息为0x00000100。

命令：0x7E 0x01 0x01 0x11 **0x02** 0x6D

响应：0x7E 0x01 0x05 0x11 0x02 0x00 0x01 0x00 0x00 0x68

3. 读软件版本（操作模式：0x03）

软件版本信息位于对象字典的0x2404 02，数据长度4字节，该值为用户设置信息。

例：如下命令读取回来的软件版本信息为0x00000100。

命令：0x7E 0x01 0x01 0x11 0x03 0x6C

响应：0x7E 0x01 0x05 0x11 0x03 0x00 0x01 0x00 0x00 0x69

4. 读产品代码（操作模式：0x04）

产品代码位于对象字典的0x2404 03，数据长度4字节，该值为用户设置信息。

例：如下命令读取回来的产品代码为0x00000001。

命令：0x7E 0x01 0x01 0x11 0x04 0x6B

响应：0x7E 0x01 0x05 0x11 0x04 0x01 0x00 0x00 0x00 0x6E

5. 读产品修订码（操作模式：0x05）

产品代码位于对象字典的0x2404 04，数据长度4字节，该值为用户设置信息。

例：如下命令读取回来的产品修订码为0x00000001。

命令：0x7E 0x01 0x01 0x11 0x05 0x6A

响应：0x7E 0x01 0x05 0x11 0x05 0x01 0x00 0x00 0x00 0x6F

6. 读产品序列号（操作模式：0x06）

产品代码位于对象字典的0x2404 05，数据长度4字节，该值为用户设置信息。

例：如下命令读取回来的产品序列号为0x00000001。

命令：0x7E 0x01 0x01 0x11 0x06 0x69

响应：0x7E 0x01 0x05 0x11 0x06 0x01 0x00 0x00 0x00 0x6C

7. 读产品名称（操作模式：0x07）

产品代码位于对象字典的0x2404 06，数据长度12字节(字符串，前11字节有效)，该值为用户设置信息。

例：如下命令读取回来的产品名称为“DeviceName”。

命令：0x7E 0x01 0x01 0x11 0x07 0x68

响应：0x7E 0x01 0x0D 0x11 0x07 0x44 0x65 0x76 0x69 0x63 0x65 0x4E 0x61 0x6D 0x65 0x00 0x84 0xFF

6.2.2 写设备信息（命令码：0x02）

写设备信息与读设备信息相对应，相同的信息都是对应于相同的对象字典。其中设备类型位于对象字典0x1000 00，因为该设备是通用设备，没有使用标准设备描述，因此该参数按照CiA定义应该为0x00000000,建议用户不要更改此参数。其操作命令及响应帧如表 57和表 58所示，其n表示该帧命令数据长度。其中命令数据的第一字节为操作模式，根据该字节判断该命令所读取的信息类别，以下将详细介绍操作命令。

注意：通过串口写入的设备信息并不能改变 XGate-COP10 设备信息，而是更改用户设备相关的设备信息。

表 57 写入设备信息命令

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x02	n	0x11	DAT	校验码

表 58 写入设备信息响应

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x02	n	0x11	DAT	校验码

1. 写设备类型(操作模式：0x01)

设备类型位于对象字典的0x1000 00，数据长度4字节，高2字节为XGate-COP10设备类型低2字节为用户设备类型代码（用户设置）。

例：写入设备类型数值为0x0011。

命令：0x7E 0x02 0x05 0x11 **0x01 0x11 0x00 0x00 0x00** 0x7E

响应：0x7E 0x02 0x01 0x11 **0x01** 0x6D

2. 写硬件版本(操作模式:0x02)

硬件版本信息位于对象字典0x2404 01，数据长度为4字节，该值为用户设置信息。

例：如下命令写入硬件版本为0x22222222。

命令：0x7E 0x02 0x01 0x11 **0x02 0x22 0x22 0x22 0x22** 0x6A

响应：0x7E 0x02 0x01 0x11 **0x02** 0x6E

3. 写软件版本（操作模式：0x03）

软件版本信息位于对象字典的0x2404 02，数据长度4字节，该值为用户设置信息。

例：如下命令写入软件版本信息为0x33333333。

命令：0x7E 0x02 0x05 0x11 **0x03 0x33 0x33 0x33 0x33** 0x6B

响应：0x7E 0x02 0x01 0x11 **0x03** 0x6F

4. 写产品代码（操作模式：0x04）

产品代码位于对象字典的0x2404 03，数据长度4字节，该值为用户设置信息。

例：如下命令写入产品代码为0x44444444。

命令：0x7E 0x02 0x05 0x11 **0x04 0x44 0x44 0x44 0x44** 0x6C

响应：0x7E 0x02 0x01 0x11 **0x04** 0x68

5. 写产品修订码（操作模式：0x05）

产品代码位于对象字典的0x2404 04，数据长度4字节，该值为用户设置信息。

例：如下命令写入产品修订码为0x55555555。

命令：0x7E 0x02 0x05 0x11 **0x05 0x55 0x55 0x55 0x55** 0x6D

响应：0x7E 0x02 0x01 0x11 **0x05** 0x69

6. 写产品序列号（操作模式：0x06）

产品代码位于对象字典的0x2404 05，数据长度4字节，该值为用户设置信息。

例：如下命令写入产品序列号为0x66666666。

命令：0x7E 0x02 0x05 0x11 **0x06 0x66 0x66 0x66 0x66** 0x6E

响应：0x7E 0x02 0x01 0x11 **0x06** 0x6A

7. 写产品名称（操作模式：0x07）

产品代码位于对象字典的0x2404 06，数据长度12字节(字符串，前10字节有效)，该值为用户设置信息。

例：如下命令写入产品名称为“XGate-Seri”。

命令：0x7E 0x02 0x0B 0x11 0x07 0x58 0x47 0x61 0x74 0x65 0x2D 0x53 0x65 0x72 0x69 0x0E

响应：0x7E 0x02 0x01 0x11 0x07 0x6B

6.2.3 写XGate-COP10 输入缓冲区数据（命令码：0x10）

XGate-COP10的输入缓冲区从编号#0~#95宽度为8bit。对应输入缓冲区16bit的#0~#47和32bit的#0~#23，它们占有相同的内存区域。如果需要对32it区的#0号区域进行操作，可操作8bit区的#0~#3以及对16bit区的#0~#1操作可得相同的结果，以此类推，因此在串口操作中只提供对8bit区域的操作。

XGate-COP10输入数据缓冲区只是一个只写区域，最多一次可写96字节的数据。操作命令和响应命令如表 59和表 60所示，在命令数据的第1字节表示当前数据在输入数据区的偏移量，所写的数据长度为n-1，

注意：偏移量+数据长度不应大于96。

表 59 写缓冲区数据命令

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)		CRC
0x7E	0x10	n	0x11	地址偏移	DAT	校验码

表 60 写缓冲区数据响应

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x10	n	0x11	DAT	校验码

例：向8bit区编号＃0开始的地址(即偏移量为0)写入8字节的数据，数据为0x12 0x12 0x12 0x12 0x12 0x12 0x12 0x12。

命令：0x7E 0x10 0x09 0x11 0x00 0x12 0x12 0x12 0x12 0x12 0x12 0x12 0x12 0x76

响应：0x7E 0x10 0x01 0x11 0x00 0x7E

6.2.4 读取XGate-COP10 输出缓冲区数据（命令码：0x11）

XGate-COP10的输出缓冲区从编号#0~#95宽度为8bit。对应输出缓冲区16bit的#0~#47和32bit的#0~#23，它们占有相同的内存区域。如果需要对32it区的#0号区域进行操作，可操作8bit区的#0~#3以及对16bit区的#0号进行操作同样对#0~#1操作可得相同的结果，以此类推，因此在串口操作中只提供对8bit区域的操作。

XGate-COP10输出数据缓冲区是一个只读区域，用户最多一次可读96字节的数据。操作命令和响应命令如表 61和表 62所示，在命令数据帧的第1字节表示当前数据在输出数据区的偏移量，第2字节表示需要读出的数据的长度。在响应命令帧数据的第1字节表示在输出缓冲区的偏移量，返回的数据长度为n-1。

注意：偏移量+数据长度不应大于96。

表 61 读缓冲区数据命令

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)		CRC
0x7E	0x10	n	0x11	地址偏移	DAT	校验码

表 62 读缓冲区数据响应

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)		CRC
0x7E	0x10	n	0x11	地址偏移	DAT	校验码

例：读取输出缓冲区偏移量从#0开始的8字节数据，所读取的数据为0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88

命令：0x7E 0x11 0x02 0x11 **0x00 0x08** 0x74

响应：0x7E 0x11 0x09 0x11 **0x00 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88** 0xFF

6.2.5 读写XGate-COP10 的NodeID（命令码：0x12）

在XGate-COP10未使DIP开关或其它方式对模块设置NodeID时，用户可以通过串口设定XGate-COP10的NodeID，设置的NodeID被保存在存储器中，模块复位后生效（设置值必须在1~127之间）。

注意：仅在未使用 DIP 开关设定 NodeID 或设定无效（0x00）的情况下，通过串口设置的值才生效。

1. 写XGate-COP10 的NodeID(操作模式：0x00)

节点NodeID位于对象字典0x2403 01，表示当前模块设定的NodeID值，设定后复位或重启模块生效。写模块的NodeID命令格式与应答命令如表 63和表 64所示，命令数据的第1字节表示当前的操作模式，第2字节表示将要写入的NodeID值。写XGate-COP10的NodeID如果成功则返回正确应答（n=0x01,命令字节第1字节为操作模式）。

表 63 写节点 NodeID 命令

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x12	2	0x11	操作模式(0x00)	DAT(NodeID)	校验码

表 64 写节点 NodeID 命令响应

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x12	n	0x11	DAT	校验码

例： 设置XGate-COP10模块的NodeID为0x20，其命令与响应如下所示。

命令：0x7E 0x12 0x02 0x11 **0x00 0x20** 0x5F

响应：0x7E 0x12 0x01 0x11 **0x00** 0x7C

2. 读XGate-COP10 的NodeID(操作模式：0x01)

读节点NodeID所读取回来的NodeID是当前节点正使用的NodeID，与写NodeID位于不同的对象字典。读NodeID位于对象字典的0x2400 00，因此写入的NodeID与读回的NodeID可能会不同。其命令模式如表 65和表 66所示。

表 65 读节点 NodeID 命令

起始字节	命令码	命令信息	特定参数	命令数据(1 Byte)	CRC
0x7E	0x12	0x01	0x11	操作模式(0x01)	校验码

表 66 读节点 NodeID 命令响应

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x12	2	0x11	操作模式(0x00)	DAT(NodeID)	校验码

注意：读回的NodeID值为当前模块正使用的NodeID，并不一定与写入的NodeID相同，需要启用写入的节点NodeID不仅需要满足3.2节所述的条件，而且模块在写入NodeID之后需要重新上电或复位才会被启用。

例：读取当前XGate-COP10的NodeID，读回的NodeID值为0x20。

命令：0x7E 0x12 0x01 0x11 **0x01** 0x7D

响应：0x7E 0x12 0x02 0x11 **0x01 0x20** 0x5E

6.2.6 读写XGate-COP10 的波特率索引值(命令码：0x13)

与设置XGate-COP10的NodeID相类似，在未使用DIP开关或其它方式设置CAN通信波特率时，用户可以通过串口来设置XGate-COP10的波特率索引值，用户所设置的波特率索引值被保存在存储器中，只有XGate-COP10上电或复位后才能生效，该值的范围为0~8，其它的值均无效。

1. 写XGate-COP10 波特率索引值(操作模式：0x00)

波特率索引值位于对象字典的0x2403 02，设定值在0~8之间有效，否则返回错误代码。其命令格式如表 67和表 68所示，当设置命令执行正确时（n=0x01,命令字节第1字节为操作模式）。波特率索引值与真实值之间的对应关系如表 5所示。

表 67　写节点波特率索引值命令

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x13	2	0x11	操作模式(0x00)	DAT(波特率索引)	校验码

表 68　写节点波特率索引值命令响应

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x13	n	0x11	DAT	校验码

例：设置XGate-COP10模块的波特率索引值为0x04，根据表 5可知，设置CAN波特率索引为0x04，则CAN通信波特率为125Kbps，命令与响应帧如下所示。

命令：0x7E 0x13 0x02 0x11 **0x00 0x04** 0x7a

响应：0x7E 0x13 0x01 0x11 **0x00** 0x7D

2. 读XGate-COP10 波特率索引值(操作模式：0x01)

读取的波特率索引位于对象字典的0x2401 00。与读NodeID相类似，读回的波特率索引值并不一定与写入的值相同，因为读回的波特率索引值为当前模块正在使用的值，只有在写入的值被启用后才会相同（模块复位或重启）。其命令帧格式如表 69和表 70所示。

表 69　读节点波特率索引值命令

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x13	1	0x11	操作模式(0x01)	校验码

表 70　读节点波特率索引值命令响应

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)	CRC

0x7E	0x13	2	0x11	操作模式(0x01)	DAT(波特率索引)	校验码

例：假设当前的模块的波特率为125Kbps，则读回的索引值为0x04，其命令与响应如下所示。

命令：0x7E 0x13 0x01 0x11 0x01 0x7C

响应：0x7E 0x13 0x02 0x11 0x01 0x04 0x7B

6.2.7 读写XGate-COP10 CAN控制器定时参数（命令码：0x14）

真实波特率是一个32位的值，包括了CAN控制器位定时以及分频参数等。与波特率索引值不一样，在XGate—COP10中已经固化了一个标准波特率表，因此可以通过索引的方式来设置CAN通信波特率，如果用户需要用到该表中没有提供的波特率，则用户可以通过命令码0x14来设置任意的波特率。可以通过由广州致远电子提供的波特率计算工具来计算波特的值，其操作界面如图 10所示，该计算软件可在http://www.embedcontrol.com/products/Can-bus/can-bus.asp中下载（文件名：“NXP CAN波特率计算软件及使用说明 ”），XGate-COP10的频率使用24MHz。当该值被设为0x00000000或0xFFFFFFFF时无效。



图 10 真实波特率计算工具

注意：在 DIP 开关设置和波特率索引值都无效的情况下，用户所设置的波特率定时参数才有效。

1. 写波特率值(操作模式：0x00)

真实波特率位于对象字典0x2403 03，只有在DIP开关设置和波特率索引值都无效的情况下才会被启用（0x00000000和0xFFFFFFFF无效），其命令格式如表 71和表 72所示，当命令执行成功则返回正确应答（n=0x01，命令字节第1字节为操作模式）。

表 71 写节点波特率值命令

起始字节	命令码	命令信息	特定参数	命令数据(5Byte)		CRC
0x7E	0x14	5	0x11	操作模式(0x00)	DAT(波特率索引)	校验码

表 72 写节点波特率命令响应

起始字节	命令码	命令信息	特定参数	命令数据(nByte)	CRC
0x7E	0x14	n	0x11	DAT	校验码

例：向XGate-COP10模块中向125Kbps的波特率值，利用图 10所示的软件计划出满足条件的值，我们选择0x2B000B这组值，其命令与响应如下所示。

命令：0x7E 0x14 0x05 0x11 **0x00 0x0B 0x00 0x2B 0x00** 0x5E

响应：0x7E 0x14 0x01 0x11 **0x00** 0x7A

2. 读波特率值(操作模式：0x01)

读波特率值与写波特率值位相同的对象字典0x2403 03，其命令模式如表 73和表 74所示。

表 73 写节点波特率值命令

起始字节	命令码	命令信息	特定参数	命令数据(1Byte)	CRC
0x7E	0x14	n(0x01)	0x11	DAT(操作模式)	校验码

表 74 写节点波特率值命令响应

起始字节	命令码	命令信息	特定参数	命令数据(5Byte)		CRC
0x7E	0x14	5	0x11	操作模式(0x01)	DAT(波特率值)	校验码

例：假设当前波特率值为0x2B000B，则读命令和响应命令如所示。

命令：0x7E 0x14 0x01 0x11 **0x01** 0x7B

响应：0x7E 0x14 0x05 0x11 **0x01 0x0B 0x00 0x2B 0x00** 0x5F

6.2.8 发送紧急代码（命令码：0x15）

当用户设备出现某种错误之后，可通过CANopen发送到CAN总线上，通知CANopen主站设备当前设备发生了错误。错误代码由5个字节组成，由用户自定义。其格式如表 75和表 76所示，当执行正确则应答正确响应（n=0x01,命令字节第1字节为操作模式）。

表 75 发送紧急错误代码命令

起始字节	命令码	命令信息	特定参数	命令数据(6 Byte)		CRC
0x7E	0x15	5	0x11	操作模式（0x00）	DAT(5 字节)	校验码

表 76 发送紧急错误代码命令响应

起始字节	命令码	命令信息	特定参数	命令数据(nByte)	CRC
0x7E	0x15	n	0x11	DAT	校验码

例：假设当前设备发生错误，错误代码定义为0x9988776655，则其命令格式如下所示。

命令：0x7E 0x15 0x06 0x11 **0x00 0x55 0x66 0x77 0x88 0x99** 0x29

响应：0x7E 0x15 0x01 0x11 **0x00** 0x7B

6.2.9 读取当前模块状态（命令码：0x16）

用户通过该命令可以读取当前模块所处的状态，所读取回来的值与当前状态的对应关系

如表 23所示。其命令格式如表 77和表 78所示。

表 77 获取当前 XGate-COP10 状态

起始字节	命令码	命令信息	特定参数	命令数据(1 Byte)	CRC
0x7E	0x15	1	0x11	操作码(0x01)	校验码

表 78 获取当前 XGate-COP10 状态响应

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x15	2	0x11	操作码(0x01)	DAT(状态值)	校验码

例：假设当前XGate-COP10模块处于操作状态，则读取的状态值为0x05。

命令：0x7E 0x16 0x01 0x11 **0x01** 0x79

响应：0x7E 0x16 0x02 0x11 **0x01 0x05** 0x7F

6.2.10 启动节点进入操作状态（命令码：0x17）

用户可通过该条命令使CANopen网络中的所有从站设备进入到操作状态，其中包括模块本身也会进入到操作状态。其命令格式如表 79和表 80所示。

注意：在网络中有主站管理的情况下，谨慎使用此条命令。

表 79 使能从站进入操作状态

起始字节	命令码	命令信息	特定参数	命令数据(1 Byte)	CRC
0x7E	0x17	1	0x11	操作码(0x00)	校验码

表 80 使能从站进入操作状态响应

起始字节	命令码	命令信息	特定参数	命令数据(1Byte)	CRC
0x7E	0x17	1	0x11	操作码(0x00)	校验码

例：使能当前CANopen网络进入到操作状态，其命令与响应如下所示。

命令：0x7E 0x17 0x01 0x11 **0x00** 0x79

响应：0x7E 0x17 0x01 0x11 **0x00** 0x79

6.2.11 改变通信串口波特率（命令码：0x18）

通信串口出厂默认波特率为115200bps,其设置范围如表 81所示。根据需求更改串口通信波特率，同时波特率索引值被保存直到再次被更改（设定波特率值时需使用当前的波特率进行通信，更改完成之后延迟一段时间再使用新设定的波特率进行通信）。禁止在应用时频繁改变通信波特率，这样可能导致通信不成功或数据丢失。

表 81 串口波特率值及索引

索引值	波特率值(bps)
0	1200
1	2400
2	4800
3	9600

4	19200
5	38400
6	57600
7	115200

其命令格式如表 82表 83所示，

表 82　使能从站进入操作状态

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x18	2	0x11	操作码(0x00)	波特率索引值	校验码

表 83　使能从站进入操作状态响应

起始字节	命令码	命令信息	特定参数	命令数据(1Byte)	CRC
0x7E	0x18	1	0x11	操作码(0x00)	校验码

例：设置当前模块的波特率为115200bps,则索引值为7，则命令帧和响应帧如下所示。

命令：0x7E 0x18 0x02 0x11 **0x00 0x07** 0x72

响应：0x7E 0x18 0x01 0x11 **0x00** 0x76

注意：建议更改完成波特率之后延长20ms左右再用新设定的波特率进行通信。

6.3　XGate-COP10 串口操作错误响应

在所有的串口操作命令中，通信过程中当命令的参数不正确或其它错误发生时，XGate-COP10都会返回错误代码（特定参数的最高位为1，表示当前为错误应答帧），错误代码帧格式如表 84所示。其中ACK与操作的命令码相同，应答操作模式码与当前命令操作模式码相同。错误代码表示当前的操作所出现错误类别，错误代码表如表 85所示。

表 84　XGate-COP10 命令执行错误响应

起始字	响应码	响应信息(长度)	特定参数	响应数据（2 字节）		校验码
0x7E	ACK=CMD	0x02	0x91	操作模式	错误代码	CRC

表 85　错误代码表

错误代码(数据)	说明	备注
0x01	命令错误	不支持该命令
0x02	数数据长度错误	超出可写区域
0x03	地址错误	超出地址范围
0x04	在操作协议栈时出现错误	-
0x05	存储数据出错	-
0x06	数据值超出范围	-
0x07	操作模式不支持	-

例：假设现在写XGate-COP10的波特率索引值为10，因为波特率索引值的范围为0~8。所以该值超出范围无效，必然在执行过程中出错，错误代码为0x06(数据值超出范围)，命令帧与响应帧如下所示。

命令：0x7E 0x13 0x02 0x11 **0x00 0x10** 0x6E

响应：7E 13 02 91 00 06 F8

附录A 串口可操作对象字典列表

操作名称	对应对象字典索引/子索引	备注
读写设备类型	0x1000 00	设备类型，(建议不用更改，非标准设备)
设备名称	0x2404 06	XGate-COP10/用户可更改(字符串,11 字节长度)，
硬件版本	0x2404 01	长度 4 字节
软件版本	0x2404 02	长度 4 字节
产品代码	0x2404 03	长度 4 字节
修订码	0x2404 04	长度 4 字节
产品代码	0x2404 05	长度 4 字节
设备名称	0x2404 06	XGate-COP10/用户可更改(字符串,11 字节长度)，
数据输入缓冲区	0x2000 00 ~0x2000 60(8bit) 0x2010 00 ~0x2010 30(16bit) 0x2020 00 ~0x2020 18(32bit)	该区域对应于 CANopen 的输入缓冲区，其中 8bit、16bit、32bit 占据相同的内存区
数据输出缓冲区	0x2100 00 ~0x2100 60(8bit) 0x2110 00 ~0x2110 30(16bit) 0x2120 00 ~0x2120 18(32bit)	该区域对应于 CANopen 的输出缓冲区，其中 8bit、16bit、32bit 占有相同的内存区
当前 CANopen 从节点号	0x2400 00	当前值
当前 CANopen 波特率	0x2401 00	当前值
模块所处的状态	0x2402 00	当前模块所处的状态
设置的 CANopen 从点 ID 值	0x2403 01	该值为节点 ID 的设置值
设置 CANopen 从节 CAN 波 特率	0x2403 02	该值为 0xFF 时无效（波特率索引值）
设置 CANopen 从节 CAN 波 特率真实值	0x2401 03	该值为 0xFFFFFFFF 时无效(波特率真实值)
通信串口波特率索引值	0x2405 00	当前通信串的波特率

附录B XGate-COP10 对象字典

索引 (Index)	子索引 (Subindex)	名称 (Name)	类型(Type)	属性 (Attr.)	默认值 (Deaf.)	描述 (Desc.)
通信参数区						
0x1000	-	Device Type	UINT32	RO	0x00000000	设备类型，高 2 字节为致远电子专用，低二字节供用户使用
0x1001		Error Register	UINT8	RO	0	当前错误类型
0x1003	0	number of errors	UINT8	RO	0	-
	1~4	standard error field	UINT32	RO	0	历史紧急错误代码
0x1005	-	COB-ID SYNC	UINT32	RW	0x80	-
0x1007		Sync Windows Length	UINT32	RW	0	-
0x1008		XGate name	STRING	Const	XGate-COP10	XGate 设备名称
0x1009		XGate hardware version	STRING	Const	V1.01	XGate 硬件版本
0x100A		XGate software version	STRING	Const	V1.00	Xgate 软件版本
0x100C		Guard Time	UINT16	RW	0	-
0x100D		Life Time Factor	UINT8	RW	0	-
0x1010	0	largest supported Sub-Index	UINT8	RO	1	-
	1	save all parameters	UINT32	RW	0	-
0x1011	0	largest supported Sub-Index	UINT8	RO	1	-
	1	restore all default para.	UINT32	RW	0	-
0x1014		COB-ID Emergency message	UINT32	RW	NodeID+0x80	-
0x1016	0	Number Of Entries	UINT8	RO	0x01	-
	1	Consumer Heartbeat Time #1	UINT32	RW	-	-
0x1017		Producer Heartbeat Time	UINT16	RW	0	-
0x1018	0	number of Entries	UINT8	RO	0x04	-
	1	Vendor ID	UINT32	RO	0x2B6	广州致远电子有限公司在 CiA 组织的厂商代码(可更改)
	2	Product code	UINT32	RO	-	XGate-COP10 产品代码

	3	Revision number	UINT32	RO	-	XGate-COP10 修订码
	4	Serial number	UINT32	RO	-	XGate-COP10 序列码
RPDO 通信参数						
0x1400	0	largest subindex supported	UINT8	RO	2	-
	1	COB-Id used	UINT32	RW	NodeID+0x200	RPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	-
0x1401	0	largest subindex supported	UINT8	RO	2	-
	1	COB-Id used	UINT32	RW	NodeID+0x300	RPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	-
0x1402	0	largest subindex supported	UINT8	RO	2	-
	1	COB-Id used	UINT32	RW	NodeID+0x400	RPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	-
0x1403	0	largest subindex supported	UINT8	RO	2	- RPDO 所使用的 COB-ID
	1	COB-Id used	UINT32	RW	NodeID+0x500	
	2	transmission type	UINT8	RW	0xFE	-
0x1404	0	largest subindex supported	UINT8	RO	2	- RPDO 所使用的 COB-ID
	1	COB-Id used	UINT32	RW	NodeID+0x500	
	2	transmission type	UINT8	RW	0xFE	-
0x1405	0	largest subindex supported	UINT8	RO	2	- RPDO 所使用的 COB-ID
	1	COB-Id used	UINT32	RW	NodeID+0x500	
	2	transmission type	UINT8	RW	0xFE	-
0x1406	0	largest subindex supported	UINT8	RO	2	- RPDO 所使用的 COB-ID
	1	COB-Id used	UINT32	RW	NodeID+0x500	
	2	transmission type	UINT8	RW	0xFE	-
0x1407	0	largest subindex supported	UINT8	RO	2	- RPDO 所使用的 COB-ID
	1	COB-Id used	UINT32	RW	NodeID+0x500	
	2	transmission type	UINT8	RW	0xFE	-
0x1408	0	largest subindex supported	UINT8	RO	2	- RPDO 所使用的 COB-ID
	1	COB-Id used	UINT32	RW	NodeID+0x500	
	2	transmission type	UINT8	RW	0xFE	-
0x1409	0	largest subindex supported	UINT8	RO	2	- RPDO 所使用的 COB-ID
	1	COB-Id used	UINT32	RW	NodeID+0x500	

	2	transmission type	UINT8	RW	0xFE	-
0x140A	0	largest subindex supported	UINT8	RO	2	-
	1	COB-Id used	UINT32	RW	NodeID+0x500	RPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	-
0x140B	0	largest subindex supported	UINT8	RO	2	-
	1	COB-Id used	UINT32	RW	NodeID+0x500	RPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	-
RPDO 映射参数						
0x1600	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21000108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21000208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21000308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21000408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21000508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21000608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21000708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21000808	映射参数
0x1601	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21000908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21000A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21000B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21000C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21000D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21000E08	映射参数
	7	PDO mapping 7.	UINT32	RW	0x21000F08	映射参数

		app. object				
	8	PDO mapping 8. app. object	UINT32	RW	0x21001008	映射参数
0x1602	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21001108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21001208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21001308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21001408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21001508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21001608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21001708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21001808	映射参数
0x1603	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21001908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21001A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21001B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21001C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21001D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21001E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21001F08	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21002008	映射参数
0x1604	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21002108	映射参数

	2	PDO mapping 2. app. object	UINT32	RW	0x21002208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21002308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21002408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21002508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21002608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21002708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21002808	映射参数
0x1605	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21002908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21002A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21002B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21002C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21002D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21002E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21002F08	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21003008	映射参数
0x1606	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21003108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21003208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21003308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21003408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21003508	映射参数

	6	PDO mapping 6. app. object	UINT32	RW	0x21003608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21003708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21003808	映射参数
0x1607	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21003908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21003A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21003B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21003C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21003D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21003E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21003F08	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21004008	映射参数
0x1608	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21004108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21004208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21004308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21004408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21004508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21004608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21004708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21004808	映射参数
0x1609	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量

	1	PDO mapping 1. app. object	UINT32	RW	0x21004908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21004A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21004B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21004C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21004D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21004E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21004F08	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21005008	映射参数
0x160A	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21005108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21005208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21005308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21005408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x21005508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21005608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21005708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21005808	映射参数
0x160B	0	number of mapped objects	UINT8	RO	8	RPDO 映射参数数量
	1	PDO mapping 1. app. object	UINT32	RW	0x21005908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x21005A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x21005B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x21005C08	映射参数

	5	PDO mapping 5. app. object	UINT32	RW	0x21005D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x21005E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x21005F08	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x21006008	映射参数
TPDO 通信参数						
0x1800	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x180	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1801	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x280	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1802	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x380	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1803	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x8000 0480	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1804	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x8000 01C0	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1805	0	largest subindex supported	UINT8	RO	0x05	-

	1	COB-ID used	UINT32	RW	NODEID+0x8000 02C0	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1806	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x8000 03C0	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1807	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x8000 04C0	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1808	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x8000 01E0	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1809	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x8000 02E0	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x180A	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x8000 03E0	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x180B	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x8000	TPDO 所使用的 COB-ID

					04E0	
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
TPDO 映射参数						
0x1A00	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20000108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20000208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20000308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20000408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20000508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20000608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20000708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20000808	映射参数
0x1A01	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20000908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20000A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20000B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20000C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20000D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20000E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20000F08	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20001008	映射参数
0x1A02	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个

	1	PDO mapping 1. app. object	UINT32	RW	0x20001108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20001208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20001308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20001408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20001508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20001608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20001708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20001808	映射参数
0x1A03	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20001908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20001A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20001B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20001C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20001D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20001E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20001F08	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20002008	映射参数
0x1A04	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20002108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20002208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20002308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20002408	映射参数

	5	PDO mapping 5. app. object	UINT32	RW	0x20002508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20002608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20002708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20002808	映射参数
0x1A05	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20002908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20002A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20002B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20002C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20002D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20002E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20002F08	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20003008	映射参数
0x1A06	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20003108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20003208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20003308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20003408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20003508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20003608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20003708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20003808	映射参数

0x1A07	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20003908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20003A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20003B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20003C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20003D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20003E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20003F08	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20004008	映射参数
0x1A08	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20004108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20004208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20004308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20004408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20004508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20004608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20004708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20004808	映射参数
0x1A09	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20004908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20004A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20004B08	映射参数

	4	PDO mapping 4. app. object	UINT32	RW	0x20004C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20004D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20004E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20004F08	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20005008	映射参数
0x1A0A	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20005108	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20005208	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20005308	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20005408	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20005508	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20005608	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20005708	映射参数
	8	PDO mapping 8. app. object	UINT32	RW	0x20005808	映射参数
0x1A0B	0	number of mapped objects	UINT8	RO	8	TPDO 映射参数数量，最大为 8 个
	1	PDO mapping 1. app. object	UINT32	RW	0x20005908	映射参数
	2	PDO mapping 2. app. object	UINT32	RW	0x20005A08	映射参数
	3	PDO mapping 3. app. object	UINT32	RW	0x20005B08	映射参数
	4	PDO mapping 4. app. object	UINT32	RW	0x20005C08	映射参数
	5	PDO mapping 5. app. object	UINT32	RW	0x20005D08	映射参数
	6	PDO mapping 6. app. object	UINT32	RW	0x20005E08	映射参数
	7	PDO mapping 7. app. object	UINT32	RW	0x20005F08	映射参数

	8	PDO mapping 8. app. object	UINT32	RW	0x20006008	映射参数
数据输入区（**TPDO** 发送数据映射区）						
0x2000	0	Number Of Entries	UINT8	RO	96	输入数据(1 字节)区的长度
	1	Application data Input 8bit #0	UINT8	RW	-	编号为#0，长度为 1 字节
	2	Application data Input 8bit #1	UINT8	RW	-	编号为#1，长度为 1 字节
	3	Application data Input 8bit #2	UINT8	RW	-	编号为#2，长度为 1 字节
	4~96	#3~#95	UINT8	RW	-	编号为#n，长度为 1 字节
0x2010	0	Number Of Entries	UINT8	RO	48	输入数据(2 字节)区的长度
	1	Application data Input 16bit #0	UINT16	RW	-	编号为#0，长度为 2 字节
	2	Application data Input 16bit #1	UINT16	RW	-	编号为#1，长度为 2 字节
	3	Application data Input 16bit #2	UINT16	RW	-	编号为#2，长度为 2 字节
	4~48	#3~#47	UINT16	RW	-	编号为#n，长度为 2 字节
0x2020	0	Number Of Entries	UINT8	RO	24	输入数据(4 字节)区的长度
	1	Application data Input 32bit #0	UINT32	RW	-	编号为#0，长度为 4 字节
	2	Application data Input 32bit #1	UINT32	RW	-	编号为#1，长度为 4 字节
	3	Application data Input 32bit #2	UINT32	RW	-	编号为#2，长度为 4 字节
	4~24	#3~#23	UINT32	RW	-	编号为#n，长度为 4 字节
数据输出区（**RPDO** 接收到的数据映射区）						
0x2100	0	Number Of Entries	UINT8	RO	96	输出数据(1 字节)区的长度
	1	Application data Output 8bit #0	UINT8	RW	-	编号为#0，长度为 1 字节
	2	Application data Output 8bit #1	UINT8	RW	-	编号为#1，长度为 1 字节
	3	Application data Output 8bit #2	UINT8	RW	-	编号为#2，长度为 1 字节
	4~96	#3~#95	UINT8	RW	-	编号为#n，长度为 1 字节
0x2110	0	Number Of Entries	UINT8	RO	48	输出数据(2 字节)区的长度
	1	Application data Output 16bit #0	UINT16	RW	-	编号为#0，长度为 2 字节
	2	Application data Output 16bit #1	UINT16	RW	-	编号为#1，长度为 2 字节
	3	Application data Output 16bit #2	UINT16	RW	-	编号为#2，长度为 2 字节

	4~48	#3~#47	UINT16	RW	-	编号为#n，长度为 4 字节
0x2120	0	Number Of Entries	UINT8	RO	24	输出数据(4 字节)区的长度
	1	Application data Output 16bit #0	UINT32	RW	-	编号为#0，长度为 4 字节
	2	Application data Output 32bit #1	UINT32	RW	-	编号为#1，长度为 4 字节
	3	Application data Output 32bit #2	UINT32	RW	-	编号为#2，长度为 4 字节
	4~24	#3~#23	UINT32	RW	-	编号为#n，长度为 4 字节
设备状态						
0x2400		Actual NodeId	UINT8		-	模块当前的 NodeID
0x2401		Actual BaudRate	UINT8		-	模块当前的波特率索引值
0x2402		ModleStatus	UINT8		-	模块当前的模块状态
0x2403	0	Entries Number	UINT8	CONST	3	-
	1	Set NodeId	UINT8	RO	-	用户通过 uart 口设置的 NodeID，可能与当前 NodeID 不同
	2	Set Index of Baudrate	UINT8	RO	-	用户通过 uart 口设置的波特率索引值，可能与当前波特率索引不同
	3	Set Baudrate Value	UINT32	RO	-	用户通过 uart 口设置的 CAN 定时参数，当该值有效，模块就会使用该值初始化 CAN 控制器
0x2404	0	Number Entries	UINT8	RO	6	-
	1	Device Hardware version	UINT32	RO	-	用户设备的硬件版本
	2	Device Software Version	UINT32	RO	-	用户设备软件版本
	3	Device Product Code	UINT32	RO	-	用户设备产品代码
	4	Device Revision Num	UINT32	RO	-	用户设备修订码
	5	Device SN Num..	UINT32	RO	-	用设备序列号
	6	Device Name	STRING	RO	-	用户设备名称
0x2405	0	UartComm. Baudrate	UINT8	RO	0x07	通信串口波特率